HDRI 3D

LIGHTING THE WAY FOR DIGITAL ARTISTS

ISSUE #3

INSIGHTS & TECHNIQUES from Professional Digital Artists

TUTORIALS IN THIS ISSUE:
MAYA • LIGHTWAVE • XSI
PYTHON • SASQUATCH
CAD MODELS

Cover Image copyright 2004 MGM Worldwide Television

ISSN 1551-689X
03
0 74470 92596 5
Issue 3 $10.95 USD $13.50 CANADA

HDRI 3D

MAGAZINE

Lighting the way for digital artists, HDRI 3D presents real world insights and emerging trends in digital production from the digital artist's perspective. Concepts and techniques are shared through production case studies and software specific tutorials by professionals in film production, architecture, advertising, medicine, and other fields. Topics are wide-ranging, covering areas such as Workflow, Visual Effects, Compositing, Lighting, Rendering, Modeling, Scripting, Animation, Character Rigging, Texturing, and more.

Go to www.hdri3d.com for more info!

ONLINE SPECIAL OFFER

☐ US $49 ☐ Rest of World $68

Subscribe to HDRI 3D Magazine
online before April 30, 2005
to get this great discount!

Six issues a year

4 EASY WAYS TO ORDER

1. In our online store at www.dmgpublishing.com
2. Fax to 1-702-616-9647
3. Phone 1-888-778-9283 or 1-702-990-8656
4. Mail to:

DMG Publishing
2756 N. Green Valley Pkwy #261
Henderson, NV 89014
USA

Name ______________________________
Address ______________________________
____
City/State ______________________________
Zip/Postal Code ______________________________
Country ______________________________
Telephone ______________________________
Email ______________________________

PAYMENT

☐ I'm enclosing a check in US Dollars payable to DMG Publishing
☐ I wish to pay by credit/debit card

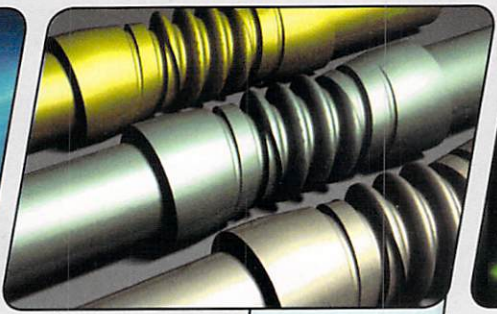
Card# ______________________ Exp Date __________
Signature ______________________ Date __________

Lighting the way for digital artists

table of contents

IN THIS ISSUE...

PAGE 6



PAGE 23

PAGE 56

departments

4 FROM THE EDITORS
BY DAN ABLAN, RAFFAEL DICKREUTER, AND DARIUSH DERAKHSHANI

82 FINAL RENDER
TRACKING THE LEARNING CURVE
BY ANDREA CARVEY

feature story

6 ALL THAT IS OLD, IS NEW AGAIN
Behind the scenes at the newest sci-fi hit *Stargate Atlantis.*
BY BRUCE G. WOLOSHYN & THE ATLANTIS VISUAL EFFECTS TEAM

articles

28 HOLY SHORTS, BATMAN!
A fast-paced animated short boasting an all-star cast of voice talent including Adam West, Mark Hamill, and Dick Van Dyke.
BY WILLIAM "PROTON" VAUGHAN

tutorials

14 LIGHTING SASQUATCH
Creating realistic lighting for Sasquatch textures.
BY NICHOLAS BOUGHEN

23 TRANSFERRING CAD MODELS TO DCC APPLICATIONS
Converting CAD models into a usable format for rendering applications.
BY JASON CLARK

tutorials

36 COMPUTER ANIMATION JOB HUNTING TIPS
Find out how to stand out from the crowd when seeking a job in this industry.
BY ED HARRISS

42 SCRIPTED RENDERING IN SOFTIMAGE|XSI
Using Python, learn tips about how to handle passes, partitions, materials, shaders, the Render Tree, and system commands.
BY BERNARD LEBEL

50 THE GRAPH EDITOR IS YOUR BEST FRIEND!
Tapping the power of the great magic wand that is the Graph Editor.
BY DEUCE BENNETT

56 DISAPPEARING/REAPPEARING ACT: ANIMATING TEXTURES IN MAYA
This tutorial demonstrates the usefulness of multiple UV sets in creating complex animated texture effects.
BY ERIC S. KELLER

64 VISUAL EFFECTS TRICKS: PORTAL PLASMA!
Visual Effects are all just a trick of the eye. Take a simple cylinder and turn it into a magical portal to another dimension.
BY ALEXANDER SHAREEF

70 SETTING UP THE HUMAN MODEL FOR ANIMATION
An easy method for rigging a biped character.
BY PETER RATNER

78 LINKING OBJECTS TO PARTICLES PART 1 - FX LINKER
Using FX Link and FX Linker to take something that is inherently difficult and make it fantastically simple.
BY ANDREA CARVEY

DAN ABLAN
EDITOR-IN-CHIEF

INFORMATION IS POWER

It seems like just the other day, Charles Edgin gave me a call and asked if I'd be interested in becoming Editor of Keyframe Magazine. Now, it's two years later and we've come a long way. HDRI 3D has not only been extremely well received by the 3D community, many people are surprised at how much they like it. All of us here at HDRI 3D thank you for your support, your ideas, and your comments. One or two of you have written in about the style of the magazine without much enthusiasm, and that's OK. Remember my last editorial? Same thing applies here – you can't please all the people all the time. Regardless, I do appreciate hearing from all of you and the publishers and I discuss each and every comment. We laugh at some of them, but we discuss them just the same! Just kidding! Seriously – thanks for emailing and keep it up.

I've been out of the loop a bit lately working with some new exciting software that I plan to cover in the magazine. It's all related to 3D and useful to your hobby or business. This new magazine format gives us the freedom to explore all the possibilities of this 3D world we work in every day. From the emails I've received, even the hardest of hardcore LightWave users are finding that reading tutorials from other applications can spark ideas. Additionally, you may find that certain applications we cover in the magazine are not something you've considered using, but by reading through a tutorial on it, you may find that it can help your workflow.

Never close any doors, especially when it comes to your creativity. Be a sponge and soak it all up. ***Information is power.***

Dan Ablan
EDITOR-IN-CHIEF

RAFFAEL DICKREUTER
SOFTIMAGE|XSI EDITOR

LATEST HAPPENINGS IN THE SOFTIMAGE|XSI COMMUNITY

For quite some time, SOFTIMAGE|XSI was one of the most expensive 3D packages on the market, and while many people were impressed by the fast workflow and the diverse toolset of the software, they argued that the package was simply out of their price range and they couldn't afford it. Some people even asked for a special "freelancer" version in the hope to get a hold of it. A few months ago at Siggraph 2004 in Los Angeles, Avid announced 3-Democracy and dropped the price dramatically. SOFTIMAGE|XSI Foundation got the new price of 495$ and became one of the cheapest packages on the market. The other versions of SOFTIMAGE|XSI, Essentials and Advanced, got through the price reduction at even a slightly lower pricing than their main competitor, Alias Maya Complete and Unlimited.

The trend that started a few years ago continued; 3D software became much more affordable, allowing literally everybody to get their hands on it.

The next surprise came along, by the end of 2004, when version 4.2 was released. Essentials started shipping with the XSI Compositor, called FX Tree, when before it had only been available in the Advanced version. XSI Advanced started to include SOFTIMAGE|Behavior, the crowd simulation software that until then was only available as a stand-alone product for 13'000$. With that, XSI Advanced became the only 3D package to include crowd simulation software.

Avid has done a lot to please its users in the past year and to increase its user base. To what extent these efforts will be successful to get more of a market share remains to be seen, but the effort is there to make the product appealing to a wide variety of 3D artists and no longer just larger production houses. With XSI Foundation, the people now can get their "freelancer" version.

A highlight product created using SOFTIMAGE|XSI is Half Life 2. When Valve's highly anticipated game was released after several months of delay last November, SOFTIMAGE released the XSI ModTool, which allows gamers to create their own characters and assets for the game. The ModTool is basically an updated version of XSI EXP, the free demo version of SOFTIMAGE|XSI. That tool might also have quite a bit of potential if it will support a broad range of game formats; however, if the amount of available exporters remains limited, the ModTool will lose popularity once the hype of a supported game starts to drop and the enthusiastic gamers move on to the next challenge that isn't supported.

The past year showed great progress for the SOFTIMAGE|XSI community as a whole. More user groups started to pop up around the world. Australia, Brazil, Japan, London, Montreal, Vancouver, Austria, and even Hawaii, as well as San Diego and New York City, all had user group events. A new user group for Los Angeles was just announced and will have its first meeting in March.

In January, a new consulting company called CG Soup was founded. Their focus is providing pipeline consulting services and custom tool development for SOFTIMAGE|XSI. The company was formed by industry veterans and SOFTIMAGE|XSI experts Kim Aldus, Bradley Gabe, and Maggie Kathwaroon. The name of the company is based on CG Supervisors, also called CG Supes.

SOFTIMAGE|XSI ARTICLES IN HDRI MAGAZINE

Scripting is a powerful tool that not only allows a user to expand the functionality of the software, but it can be big a big timesaver for repetitive tasks. Since SOFTIMAGE|XSI supports several different scripting languages, such as Jscript, VBScript, PerlScript, and Python, it gives the user the choice to choose the language, which is especially convenient for experienced programmers that can jump right into scripting if they know one of these languages. However, if a beginner intends to start scripting with, for example, Python, difficulties will arise quickly, as the documentation contains mostly only examples of VBScript and Jscript. With limited documentation, the user might have a hard time making progress. This issue of HDRI Magazine tries to address this problem a bit. Bernard Lebel will explain how to use Python to handle passes, partitions, shaders, and the render tree in his article "Scripted Rendering With SOFTIMAGE|XSI."

Ed Harriss, industry veteran and XSI expert, focuses his article on job hunting, which has become more and more difficult, since every year so many schools train 3D artists and output new talent that are competing for the available jobs. Also, with the training available today, such as videos, books, and Web tutorials, many people are self-taught and try to break into the industry. So, being good at 3D is only one part of the business; the other is how well you can sell yourself. Since Ed wrote an entire book on that topic, he can give great insight into this area in his article.

The goal for future issues of the SOFTIMAGE|XSI section in HDRI Magazine will also be to touch topics that haven't been covered yet in other magazines or on the Web, and help to fill in gaps so that users can learn about the potentials of SOFTIMAGE|XSI and experiment with features they didn't touch before.

DARIUSH DERAKHSHANI
MAYA EDITOR

KOOSHLINGS

My friends call me Koosh. It came about from a strange series of events way back in high school, but it's stuck ever since. It's my pleasure to come on as HDRI 3D's the new Maya Editor with this, the third issue to date. It's nice to see a publication dedicated to bringing together the different trains of thought under one cover, and as we progress, I hope to help make HDRI 3D a very influential component of the professional and academic CG landscape. It's amazing how free and open people are to sharing their knowledge to help out students and colleagues, and it is with this spirit of open communication and the energy of software makers like Alias that we will all expand and mature this exciting field. I look forward to looking forward, and to kick off this Kooshlings Column with some of my thoughts of the future: There is a powerful interest in CG creeping into the mainstream, but how do we help nurture it further?

"3D KLEENEX"

Like the brand name Kleenex has overcome the actual product's name (tissues) in people's way of speaking, the term "Photoshop" is now as much a verb as a brand name, and has come to represent image editing for many people in the industry and consumer market alike. This is in no small part due to the deluge of inexpensive digital cameras and their image editing applications in the consumer marketplace. It would be rather difficult to find a person who would not know what someone means with the phrase "I Photoshopped myself into that picture."

So when do we see a similar treatment with 3D? At what point would you be comfortable saying, "I Maya-ed a spaceship into my son's birthday party video" to your technophobic, if not barely literate, neighbor and not get looked at as if you stepped on his hamster?

It's safe to say that 3D software has indeed gone through a massive move into the enthusiast market. With Moore's Law observing that CPU power roughly doubles every 18 months, a computer on your kitchen counter is very capable of handling 3D content creation, as well as digital video editing and compositing. It's refreshing to see this trend create excitement for a lot of computer and design enthusiasts looking for a new creative outlet for themselves. Add to that rigorous price drops for these computers and the 3D applications, as well as moves by some of the software makers to create and distribute free learning editions of their applications—such as Alias' Maya Personal Learning Edition—and you have an enthusiast market that is, well, enthusiastic. This also proliferates 3D well beyond the typical digital content creation industries into the hands of normal people. Well, as normal as people like us can get.

But what's it really going to take to get 3D software like Maya into the consumer mindset, still one critical step from the enthusiast mindset? Enthusiasts, to some extent similar to pros, don't mind paying a premium for good hardware, and they'll spend the time and money learning the complicated applications. When do we see a good push into the consumer market? When can the average Joe pick up his mouse and animate a talking sandwich into digital video of his weekend trip up to the cabin to see his other wife and the kids no one knows about?

Certainly the technical requirements are there in even the mid-level consumer computer, so the machines have ability, but what about their users? Simply, there is a level of intimidation here for them that keeps 3D out of the common market, so that we don't currently see 3D applications on the shelf at Best Buy. 3D applications, for the most part, are just too complicated to learn for the average headed-to-an-early-heart-attack Best Buy browsing kind of guy.

Not to insinuate that Best Buy gives you heart disease or that Photoshop is a snap to learn, but because this average person can pick up a digital camera and transfer his images to his PC as easily as heating up a burrito, he will eventually be faced with the option, perhaps the delicious lure, of Photoshopping these images at some point, all in due course. Little red eye editing here and there and before long, you have this guy's illegitimate kids flying through pictures of clouds pilfered off the Internet.

So either we need to make everyone attend an art school that teaches 3D or come to a point where 3D applications are "point and shoot" easy to operate, which could mean cutting out a lot of the features. But is this a bad thing, per se? If you take a look at the feature set behind an inexpensive image editor bundled with a point and shoot digital camera and the full-featured Photoshop, you will see a big difference in what you can do. This distinction can simply be made with the line between amateur and professional. The pro needs a set of tools to satisfy the endlessly needy art director who "just can't quite put his finger on it," while the average consumer will just need to fix a few image flaws and maybe cut out an ex-boyfriend from a vacation snapshot.

With a lighter version of Maya, for example, the consumer can be given just enough tools to model and animate a simple talking sandwich, or dancing peppers to liven up his lunchtime DV. Perhaps add-in modules can expand upon that capability, much like Photoshop plug-ins, and give this person the chance to animate particle steam from the dancing peppers, or create reflections and full-on characters.

Whatever the exact specs, a lighter featured and therefore easier to use 3D package is key for the basic consumer market to enjoy 3D creation. There are a few of these offerings out now, like Carrara Basic or Poser or Animation Master, but they have not yet set fire to the consumer market place. Perhaps big name recognition of a "Maya Limited Edition" would lend more credence to a consumer movement?

In any event however, with DV and firewire becoming the norm for most people with video cameras, consumers will eventually see the many options, even the allure of adding animated 3D elements to their videos as much as enthusiasts are seeing today. Giving them an affordable and fairly featured but less complicated option becomes perhaps the difficult onus of the software makers to integrate creating easier CG with a typical family's captured DV footage. Teenage Johnnie can sit down at the family PC and pound out a CG spaceship using a simplified GUI in such an application, keyframe it flying around in the background of their summer trip to visit the uncle no one talks about, and tell his friends he "Maya-ed it in there."

The thing is, until software makers take a bigger leap of faith into this market with such an offering, consumers will continue to forgo 3D as a common toolset for their everyday computing. Perhaps with the soon-to-come truly 3D-space desktops in an OS like Windows, the consumer market will soften up more to desktop CG; giving people more time to adjust to this new CG lexicon and workflow may have to be the key. At some point, though, we eventually have to see a stripped down Maya or LightWave or SOFTIMAGE|XSI that can easily integrate CG into family videos at a consumer software price. Of course, this assumes the task of making such an application from the robust and deep software in production environments now is anything less than excruciating for a company to undertake. This point will come, and if Moore's Law is to apply to how deeply people use their computers as well as their CPU speeds, then the alarm is about to go off for 3D.

HDRI 3D
Issue #3

PUBLISHER *Alice M. Edgin*	alice@dmgpublishing.com
EDITORIAL DIRECTOR *Charles A. Edgin*	charles@dmgpublishing.com
EDITOR-IN-CHIEF *Dan Ablan*	dan@hdri3d.com
MAYA EDITOR *Dariush Derakhshani*	dariush@hdri3d.com
SOFTIMAGE|XSI EDITOR *Raffael Dickreuter*	raffael@hdri3d.com
TECHNICAL EDITOR *Brad Carvey*	bradcarvey@hotmail.com
MANAGING EDITOR *Amber Goddard*	amberg@hdri3d.com
CONTRIBUTORS *Deuce Bennett*	deuce@cimagineering.com
Nicholas Boughen	nick@rainmaker.com
Andrea Carvey	andreacarvey@hotmail.com
Jason Clark	jason@jase.ca
Ed Harriss	edharriss@edharriss.com
Eric S. Keller	kellerrific@yahoo.com
Bernard Lebel	bernard@bernardlebel.com
Peter Ratner	ratnerpj@jmu.edu
Alexander Shareef	shareef@dx-digital.com
William 'Proton' Vaughan	Proton@spinquad.com
Bruce Woloshyn	bruce@rainmaker.com
ART DIRECTOR *David S. Wilkinson*	david@sabredesign.net
LAYOUT/DESIGN *Sabre Design*	artists@sabredesign.net

HDRI 3D MAGAZINE ADVISORY BOARD
Timothy Albee • Akiko Ashley • Jack "Deuce" Bennett II
Stephen M. Burns • Kevin Cahill • Dwayne J. Ferguson
Todd Grimes • Ed Harriss • Varto Keshishian
Policarpo • William Vaughan

ADVERTISING SALES
Charles A. Edgin charles@dmgpublishing.com
1-702-990-8656

SUBSCRIPTIONS & BACK ISSUES
Order online at www.dmgpublishing.com
Email: info@dmgpublishing.com
1-702-990-8656 or 1-888-778-9283

FOR ADDRESS CHANGES
Email: moving@hdri3d.com
Via Mail: DMG Publishing
2756 North Green Valley Pkwy #261
Henderson NV 89014

HDRI 3D MAGAZINE
(ISSN 1551-689X)
is published bimonthly by
DMG Publishing
2756 North Green Valley Pkwy #261
Henderson NV 89014
USA

© 2005 by DMG Publishing. All Rights Reserved. No part of this publication can be reproduced or transmitted in any form or by any means, electronic or mechanical including photocopying, recording, or information storage and retrieval system without permission in writing from the publisher.

DMG Publishing specializes in the publication of high-quality magazines for the digital visual effects community. As a 2002, 2003 and 2004 Graphics Excellence Award winner, DMG's mission is to be responsive to community needs and to draw together the finest to create relevant, superior, and vibrant publications using the latest printing technologies. With a world-wide distribution of its magazines, DMG prides itself on high standards of customer service and subscriber support.

"ALL THAT IS OLD, IS NEW AGAIN"

BY BRUCE G. WOLOSHYN & THE ATLANTIS VISUAL EFFECTS TEAM

copyright 2004 MGM Worldwide Televi

STARGATE
ATLANTIS

copyright 2004 MGM Worldwide Television

copyright 2004 MGM Worldwide Televi

TOP, THE LOST CITY OF ATLANTIS
LOWER LEFT, OPENING TITLE CARD
LOWER RIGHT, PUDDLE JUMPER IN FLIGHT

Stargate Atlantis is MGM Television and the Sci-Fi Network's new spin-off series, based on one of science fiction's most exciting television shows, *Stargate SG-1*. It follows an entirely new Stargate team as they embark on a dangerous mission to the distant Pegasus Galaxy, where they discover the mythical lost city of Atlantis ... and a deadly new enemy, the Wraith.

When I was first contacted by Rainmaker president, Bob Scarabelli and told that we had been awarded the visual effects for the series pilot, "Rising," I was elated. We had been involved in the creation of the visual effects for *Stargate SG-1* since its pilot back in 1997, and now we would have the opportunity to help design an entirely new version of one of the genre's most popular shows. Little did I realize exactly how much work this project was to entail.

A NEW PIPELINE

First, we wouldn't just be working on the pilot to the new series. As a matter of fact, to adhere to the new show's production schedule, we would actually be working on (and delivering) several subsequent episodes of *Stargate Atlantis,* even as we were still designing and creating effects for the pilot. In addition, we would still be contributing effects to season eight of *Stargate SG-1*. Plus, just to make things a little more interesting, both of the shows would no longer be shooting on film and finishing in standard definition as we had for the previous seven seasons on *SG-1*. Both *Stargate SG-1* and *Stargate Atlantis* would be making the jump to high definition video acquisition and finish in full 24p HDTV.

This new acquisition and delivery standard was one of the first challenges that we would have to incorporate into our new production pipeline for the show. From the visual effects standpoint, this would lead to a list of concerns regarding infrastructure, system upgrades, staffing, rendering, and delivery schedules. Clearly, the high definition hurdle would be one of the first things we would have to clear in preparing to start work on the show.

For both *Stargate Atlantis* and *Stargate SG-1*, the answer would lie in producing the visual effects in a combination of full HD, as well as SD. We would then up-convert the finished SD effects back to HD for insertion into the HD-assembled show. Fortunately, digital developments at Rainmaker have made this a seamless integration, as the SD visual effects maintain rigid adherence to the progressive video format. That is to say, the visual effects are all executed at 24 progressive frames per second, even though the spatial resolution is standard definition.

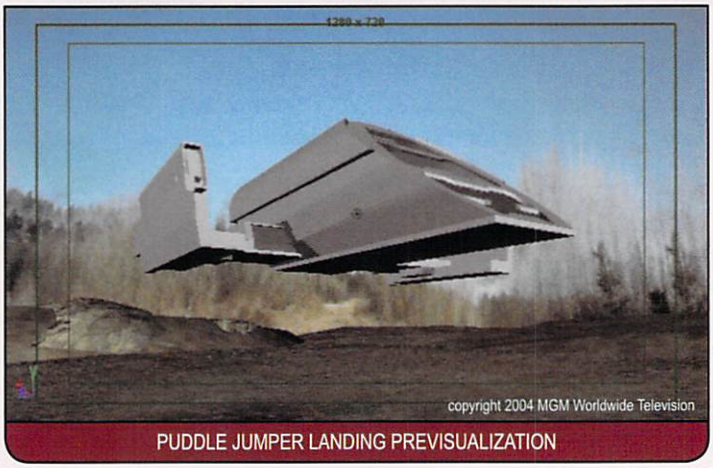
For season eight of *Stargate SG-1* and the inaugural season of *Stargate Atlantis*, this HD-SD-HD visual effects pipeline has proven to be a robust performer, with the final HD tape proceeding fluidly into and out of the VFX shots. In a perfect world model, of course, everyone would prefer all of the visual effects to be native HD resolution. However, with the size, scope, and schedule of the visual effects on both series, the transcoded SD-HD hybrids are proving to be the best of both worlds.

PREVISUALIZATION

With the *Stargate Atlantis* pilot's schedule overlapping so many other episodes of both series, the efficiency of shot-by-shot execution was always at the forefront of our planning. To this end, we utilized previsualization to a much greater extent than we had ever done before on *SG-1*. In many cases, our final shots matched second-for-second to the previs that was approved before the cut of the pilot was even locked.

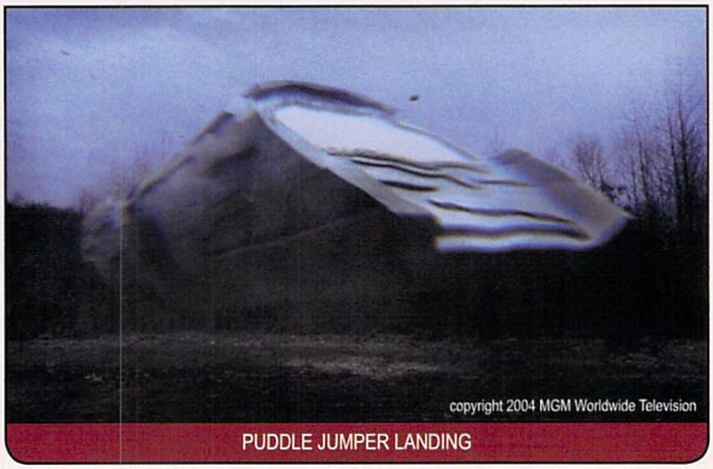
This production model not only helped us to lock a significant amount of visual effects shots before editorial locked the cut of a particular scene, but sometimes even before we went to camera. Completing previs prior to going to camera then allowed us to further refine our elements list so that we knew exactly what we were shooting on the day. In this way, many times our artists were following an already producer-approved template, focusing on the art of their discipline, rather than wasting time working out the blocking.

There is no better example of this process than the climactic space battle at the end of the show. The sequence consisted of over 30 shots, all of which had only brief descriptions of the action in the script. In order to better work with the producers and director in choreographing the battle, we set up an on-site animation station. This physical proximity to production allowed us to be able to receive quick feedback and implement changes much faster. Because the shots were prevised in LightWave, we were able to come back to Rainmaker, while

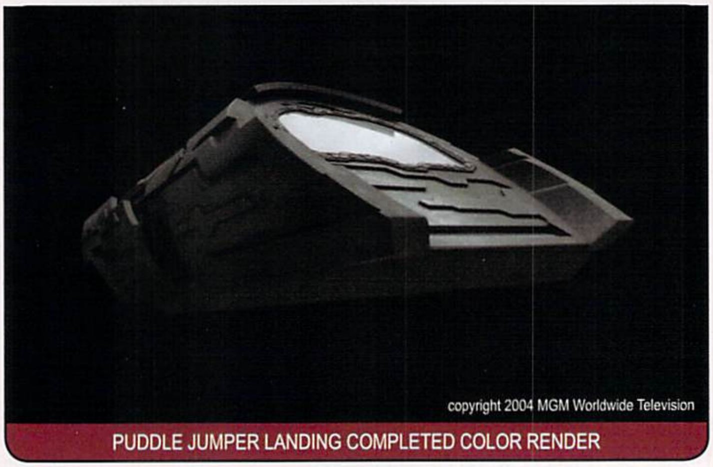


copyright 2004 MGM Worldwide Television

PUDDLE JUMPER LANDING PREVISUALIZATION


copyright 2004 MGM Worldwide Television

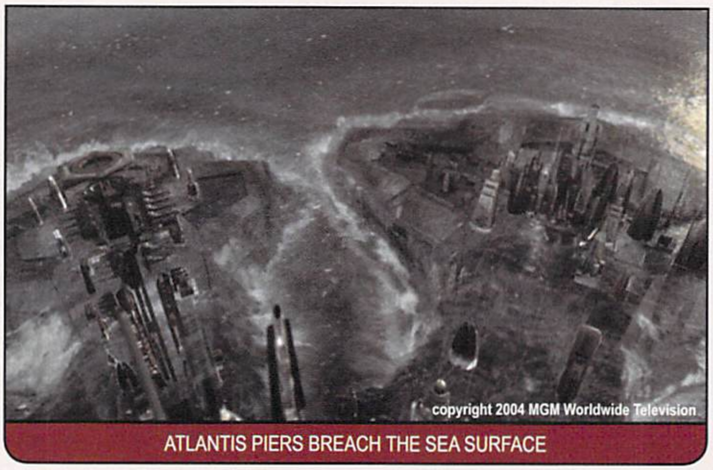
PUDDLE JUMPER LANDING


copyright 2004 MGM Worldwide Television

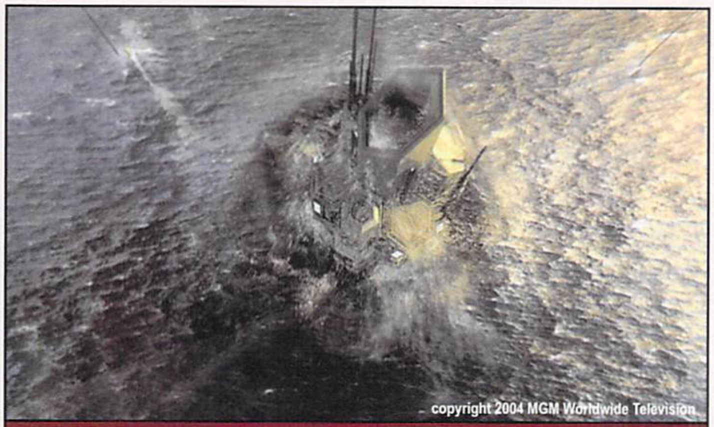
PUDDLE JUMPER LANDING COMPLETED COLOR RENDER

copyright 2004 MGM Worldwide Television

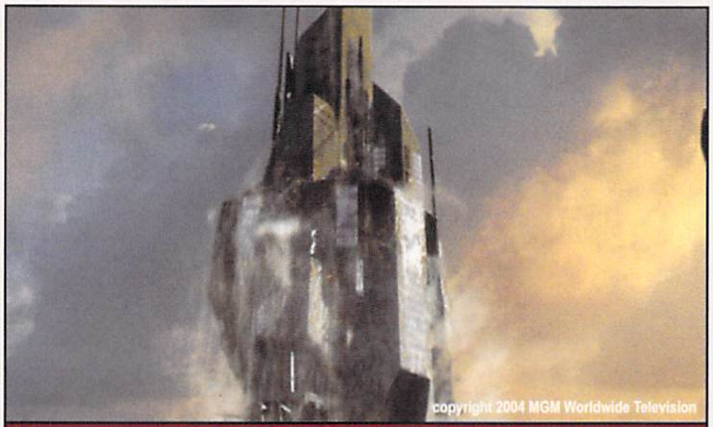
ATLANTIS PIER E LARGE-SCALE STEEL BLACK BOX MODEL


copyright 2004 MGM Worldwide Television

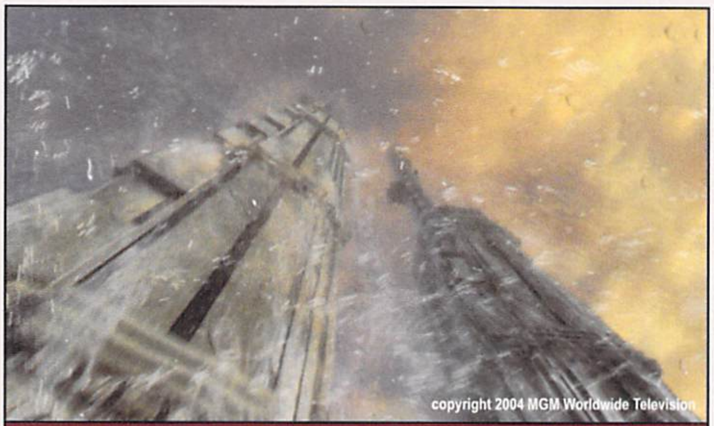
ATLANTIS PIERS BREACH THE SEA SURFACE


copyright 2004 MGM Worldwide Television

ATLANTIS CENTRAL TOWER BREAKS THE SURFACE


copyright 2004 MGM Worldwide Television

ATLANTIS CENTRAL TOWER WITH MAYA PARTICLE ANIMATION


copyright 2004 MGM Worldwide Television

AS THE CAMERA BREAKS THE OCEAN SURFACE

the animation was still being created, and substitute the previs models with the high-resolution ships that were being built. Animation and camera movement could be approved with the low-resolution models and then sent back to have the real ships substituted in their place. This provided a seamless integration from previs to final renders that allowed us to get the high volume of shots in a very short timeframe.

MORE THAN JUST CG

One of things that I think is kind of sad in today's world of visual effects is that everyone thinks, "They just did that with a computer." In the world of both *SG-1* and *Atlantis*, this couldn't be further from the truth. As a matter of fact, the elements shoot for the *Atlantis* pilot was the most comprehensive elements shoot ever for *Stargate*.

For instance, even though we were creating literally hundreds of passes of particles in Maya for the sequence where the city of Atlantis rises up to the surface of the ocean, we also knew that nothing looks more like water, than ... well ... real water. To facilitate some of the huge scale water effects we needed as the city's massive pier decks breeched the surface, we turned to Wray Douglas and the special effects department. Working to a scale drawing of the LightWave model of Atlantis's pier E, the special effects department created a large-scale model of the pier out of industrial-strength steel. This black box model, over seven meters in length and weighing in at a little over two tonnes, would represent the virtual 625 meter pier as it broke the surface of the ocean. The massive steel model was then hauled to Vancouver's False Creek, and repeatedly dropped into and lifted from the water by a huge construction crane. Filmed at 120 frames per second from a variety of angles, this large-scale model provided us with a wealth of real world water elements to composite into city rising sequence.

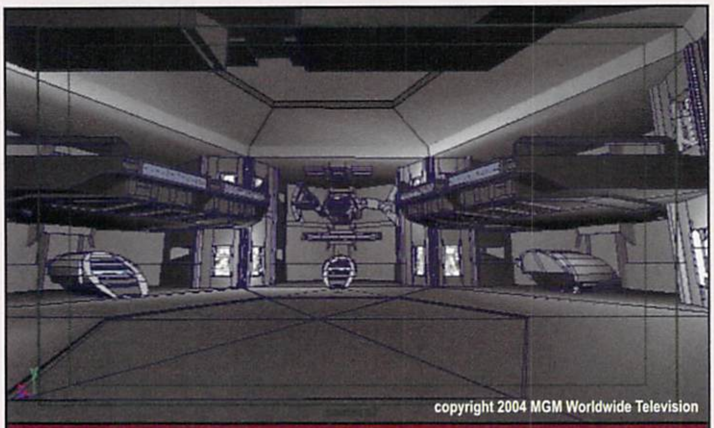
To create the Wraith Phantoms, a hallucination projected by the Wraith creatures, we needed a quick and dirty technique that would give us lots of coverage and angles that we could not only use on the pilot, but also re-use for future yet-to-be-written episodes. To do this, the wardrobe department made us a costume of streamers in black, white and blue. We then shot on a bluescreen stage, so that wherever there was blue fabric on the Phantom it would be transparent. Adding fans blowing at the actor, the streamers would blow up into the air. Taking that footage into the Inferno workstation and averaging it, prior to compositing, gave the creature the ghostly feeling the producers were looking for.

copyright 2004 MGM Worldwide Television

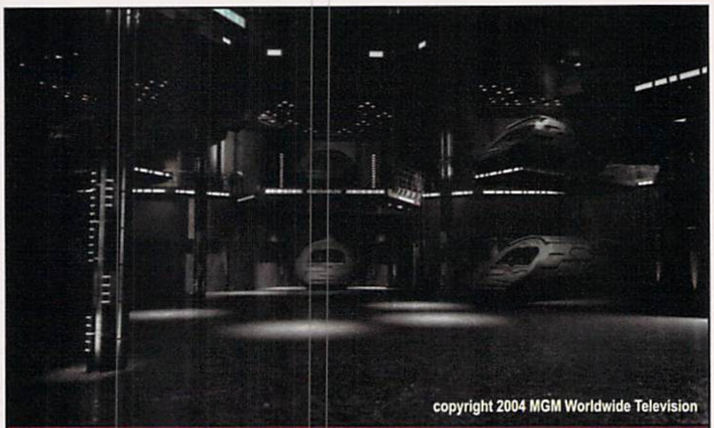
LIGHTWAVE OCEAN SURFACE AND MAYA PARTICLE SPRAY

copyright 2004 MGM Worldwide Television

ATLANTIS OPENING SHOT


copyright 2004 MGM Worldwide Television

PUDDLE JUMPER BAY INITIAL DESIGN MODEL


copyright 2004 MGM Worldwide Television

PUDDLE JUMPER BAY FINAL COLOR RENDER

Finally, to complete our elements shooting, we shot for a day in the parking lot at the studio. Our shot list for this day included a lot of scene-specific elements that we would need, based on the design work that we had already completed in the digital realm. Once again, stressing that actual photography of elements would assist us just as much as our digitally created ones. Filming at a variety of frame rates, we shot water being splashed onto a sheet of glass for the shot where the camera breaks the surface of the water. We also shot a great many plates of general water splashing down the sides of black box surfaces. These would then be used as texture maps on the LightWave model to help make our city look wet. Of course, as with all of our other elements, we would multi-pass render a variety of "wet look" texture maps for maximum control by our compositing team.

MODELING, LOTS OF MODELING

The one thing that really differentiated the Atlantis pilot, "Rising," from working on SG-1, was the sheer amount of 3D modeling that had to be done. Every single model we needed, from the digital props and set pieces, to the spacecraft and city itself, had to be modeled from scratch. There would be no library of past seasons to draw on for this show.

"Rising" begins as we see an alien spacecraft, called a Puddle Jumper, flying over Antarctica towards the lost city. The flight path was carefully planned, but covered an immense area. Detail in the icy plain was built up in the 3D model so that we only saw exactly what was in the camera view; zones of priority were set depending on how close the camera would actually get to any given feature. The snow and ice surface used a great many procedural textures since they work well on a micro and macro scale. Maya fluids were also utilized in the opening sequence to push a wall of snow and ice towards the camera as Atlantis leaves Antarctica.

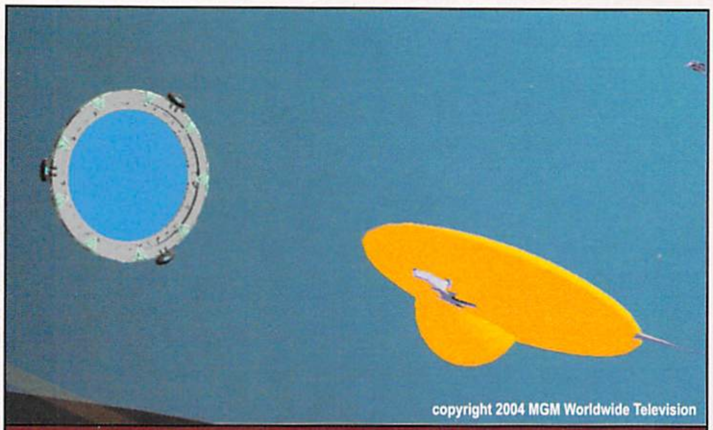
The Puddle Jumper itself was first built as a practical set piece that we would then replicate in CG. At first glance, the Puddle Jumper seems like a very simple cylinder, but with all the complex curves built into the paneling, it proved a little more challenging to recreate. We took digital photographs of the practical set piece that we used for texture maps. We then removed as much lens distortion and irregular lighting as we could, and stitched together the top, front, back, and side angle views. We then took those assembled templates, stretched them to conform to our on-set measurements, and ended up with photo-real blueprints from which we could build the Puddle Jumper. This made the texturing phase

copyright 2004 MGM Worldwide Television

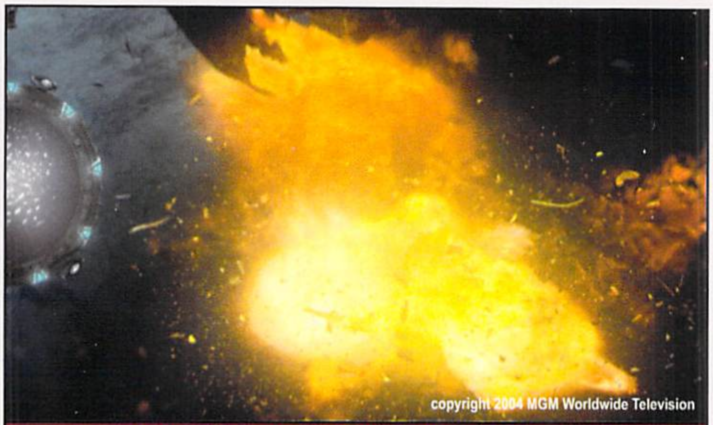
THE LOST CITY OF ATLANTIS

copyright 2004 MGM Worldwide Television

A PEGASUS GALAXY STARGATE IN ORBIT


copyright 2004 MGM Worldwide Television

SPACE BATTLE EXPLODING DART PREVISUALIZATION


copyright 2004 MGM Worldwide Television

COMPLETED LIGHTWAVE EXPLODING DART FIGHTER

of the model a snap, because everything lined up perfectly.

Due to the hectic shooting schedule, we actually had only about a 15-minute window on-set to photograph the interior of the practical Puddle Jumper cockpit. However, from these photographs, our digital interior modeling ended up being fairly simple, and there wasn't really a whole lot of geometry needed, because the photographs that we used as texture maps were so good. In some cases, we can actually move the camera right up to a dial on the main console and everything still looks great. We later took photographs of the actors, mapped them onto flat planes, and dropped them inside the cockpit model. A very simple trick, but we had some people scratching their heads trying to figure out how we got those guys in there. Then, after getting the majority of the 3D model to exactly match the practical set piece; the engines and their respective compartments, missile bays, and the underside of the Puddle Jumper; were designed and modeled in LightWave to match the style of the rest of the ship.

We followed a similar modeling scheme for many of the CG models that had to match existing set pieces, including the Atlantis Stargate; taking photographs and stretching them to fit our measurements, which we then used as modeling templates.

As well as creating props and ships, we were also faced with creating several digital set extensions and virtual environments. The Puddle Jumper bay, located in the central tower of the city, is a good example. The Puddle Jumper bay is basically a parking garage and maintenance facility for up to twelve Puddle Jumpers, located directly above the city's main gate room. The starting point for this was the practical set piece at the studio. What was practically built represented the rear of one of the six bays on the main floor of the two-story structure. Texture photographs were taken of all of the on-set walls. These photographs were later digitally painted into more variations for the many more digital walls. While still at the studio, we then trekked over to the main Gaterium set, and in between the live action takes, we took additional photographs. These would provide additional textures and act as reference for the architectural style favored by the ancient builders of Atlantis. This was to prove very important, as we constantly strived to make the digital sets look like they belonged with all of the live action sets.

Lighting the bay was done with 200 lights of all types: shadow mapped spots, point lights, and area lights. To make rendering more efficient, some smaller light sources were baked onto the actual textures. A few radiosity test renders were done to get a feel for the look we were after. This look was then reproduced using regular lighting to, once again, keep our rendering times reasonable. Getting shots out fast meant building only the sections visible in the first shots, then finishing off the rest of the bay as time permitted.

The centerpiece for the pilot's modeling was, of course, the exterior of the lost city itself. The exterior Atlantis model went through many design iterations, and by the time it was completed, it came in at around four million polygons and over a gigabyte of textures. Because of its size and the variety of shots present in the pilot, it needed to look good from all angles; hence, it was an extremely complex model. In lighting the city, we created a lighting rig with a combination of directional and area lights that could be adapted to unique situations, but which allowed us to keep consistency through all of the shots, not to mention show off all the work that had been done by the modeling team on this monster model.

DIGITAL WATER

Although we had a massive list of models that needed to be created for the pilot, none was more daunting than the modeling of the surface of the sea surrounding the city of Atlantis. We struggled at first whether to go with Maya, because of its inherent water controls, or LightWave, as that's what we would be creating the city and all of the master scene files with. In the end, we ended up going with both. The main water surface was created in LightWave and utilized the NatureFX plug-in from Dynamic Realities to help finesse the look. We then used Maya for the hundreds of layers of particles and splash dynamic simulations.

CITY RISING

In the city rising sequence, we had to push this four million-polygon model through the ocean surface and get a convincing "breaching" of the water. This meant developing a pipeline to get scenes back and forth from LightWave to Maya, as LightWave was used for the rendering of all primary elements (water, city and sky), while Maya was used for particles and splash dynamic simulations. Simulation was also used in LightWave to get the water to bulge and push out as the buildings passed through the surface.

In order to give the illusion of water dripping and falling down the sides of the buildings in the rising sequence, we did a great deal of multi-pass rendering to provide our compositing team with many layers of the buildings to achieve the proper look. RGB, z-depth, specular, and shading passes for the city, as well as various water falling elements that were both used from our elements shoot and textured onto the buildings, as well as custom procedurals.

SPACE BATTLE

Our theory with respect to lighting the space battle sequence is similar to how we like to design spacecraft. That is, to always try to make the ships look cool from any angle, even if you have to sacrifice a little bit of realism. Unfortunately, if you have to light a planet from one angle and then you do a reverse of that shot in the very next cut, the lighting has to match. But we try to cheat as much as possible if it's going to make a shot look better or more dynamic. Generally, wherever you go in 3D space, strong backlighting can help heighten the dramatic effect. Blue or orange fill always looks great, especially with the evil Wraith Dart fighters. Also, if ever there is a question about where that light source came from, thank goodness for the off-camera stars that wander through space.

One of the many technical challenges we faced in the *Atlantis* pilot was getting the Wraith Dart ailerons to rotate automatically based on the attitude of the Dart. The goal was to eliminate the need to hand-animate the ailerons, so the artist need only turn and bank the ship. Of course, anyone who has observed aircraft knows that ailerons will move before the craft begins to turn, so we needed a way to anticipate what the craft was going to do before it happened, and feed that information to the aileron rotation nodes.

This was done first by adding a null object to the Dart. This object (which is unseen in the shot) simply records the motion path and rotation values of the Dart. Using mathematical expressions, we were able to look at the orientation of that Null object at any time in the shot. We would then take the Null's orientation at the time (NOW), and compared it with the Null's orientation at the future time (30 frames from NOW), taking the difference between the two as the rotation value that was applied to the ailerons. One aileron was given this as a positive value, and one as a negative value, since the ailerons rotate opposite each other in order to make the craft bank. We also had to apply rotation limits to the ailerons, otherwise they would just spin around and around if the craft performed a barrel roll. Once the expression and limits were applied, we had a craft whose ailerons would rotate automatically in anticipation of an upcoming turn.

We also wanted the Darts to explode more dramatically than just being covered by a single 'poof' explosion. To that end, we created a "scored" stunt Dart, much the same way that productions have stunt cars. The hull of the stunt Dart was given three-dimensional thickness and divided up into a couple dozen areas, which were in turn broken into tiny pieces. All or part of any given area could be blown apart in large chunks or tiny pieces with independent explosion simulation timing and magnitude for each given area. This allowed us to start (and end) a Dart's destruction based on where it was actually hit by blasts and to make the Dart's annihilation last to fit any given shot.

PEOPLE MAKE PICTURES

As I write these words, we have just completed the final shot on the last episode of season one of *Stargate Atlantis*. I find it hard to write about this newest incarnation of the series without feeling a profound sense of pride about what our team has accomplished. I've always felt very strongly about stressing that although we live in an age where incredible advancements in technology have allowed us unprecedented control in the images we create, that it is the "people" that create these images, not the technology. This year, the first of our new journey to Atlantis and the Pegasus Galaxy, I have been honored to work with some of the finest artists and technicians I have ever known. Everyone made sacrifices to deal with the almost impossible schedule, everyone brought all the techniques they knew to the table, and most of all… everyone cared. To each one of them, I extend my sincere thanks for a job well done. No, superbly done.

I'd also like to thank our executive producers, Brad Wright and Robert C. Cooper, for entrusting us with their world and the characters that inhabit it. In a year that has seen two such huge visual effects projects on the go simultaneously, and particularly the pilot to *Stargate Atlantis*, their guidance and support, and that of visual effects producer Michelle Comens and director Martin Wood, has made an immeasurable difference.

With *Stargate Atlantis* premiering on the Sci-Fi Network to more than 4.2 million viewers, "Rising" earned a 3.2 Nielsen rating. This set a new ratings record on the Sci-Fi Network, and should help to keep us busy exploring the Pegasus Galaxy for some time to come.

STARGATE ATLANTIS PILOT "RISING" VISUAL EFFECTS CREDITS

Michelle Comens	Visual Effects Producer
John Gajdecki	Visual Effects Supervisor
Bob Scarabelli	Rainmaker Visual Effects Executive Producer
Bruce G. Woloshyn	Digital Effects Supervisor & Lead Digital Compositing Artist
Jinnie Pak	Visual Effects Co-ordinator
Tara Conley	Visual Effects Co-ordinator
James Rorick	Visual Effects Co-ordinator
Janice Groom	Visual Effects Operations Supervisor
Dan Mayer	Lead 3D Animator
Wes Sargent	Lead 3D Animator
Rod Bland	3D Animator
Jose Burgos	3D Animator
Nicholas Boughen	3D Animator
Tom Brydon	3D Animator
Craig Calvert	3D Animator
Ho Sung Cheon	3D Animator
Bryan Davies	3D Animator
Tristam Gieni	3D Animator
Trevor Harder	3D Modeller
Sean King	3D Animator
Megan Majewski	3D Animator
Krista McLean	3D Animator
Daniel Osaki	3D Animator
Mark Pullyblank	3D Animator
Les Quinn	3D Animator
Vishal Anand	3D Animator
Debora Dunphy	Lead Digital Compositing Artist
Simon Ager	Digital Compositing Artist
Jordan Benwick	Digital Compositing Artist
Kristy Dearholt	Digital Compositing Artist
Chris Doll	Digital Compositing Artist
Keegan Douglas	Digital Compositing Artist
Peter Hunt	Digital Compositing Artist
Mathew Krentz	Digital Compositing Artist
Colin Liggett	Digital Compositing Artist
Tannis Mathers	Digital Compositing Artist
Christine Petrov	Digital Compositing Artist
Lee Pierce	Digital Compositing Artist
Carmen Pollard	Digital Compositing Artist
Gary Poole	Digital Compositing Artist
Trevor Strand	Digital Compositing Artist
Chris Wren	Concept Artist
Tracey Baxter	Rotoscope Artist
Madhava Reddy	Rotoscope Artist
Arnold Yuki	Rotoscope Artist
Hamish Hamilton	Visual Effects Assistant
Ken Hayward	Director of Technical Development
Ronald Knol	VFX & IT Engineering Supervisor
Zane Harker	Resource Manager
Joe De Michelis	Information Systems Administrator
Chi Pham	VFX Systems Administrator
Grant Bowen	Applications Support Specialist

In Loving Memory of

Bob Scarabelli

1955 - 2004

Rainmaker Founder,
Mentor and Friend

A man whose character and vision will always inspire me.

BRUCE G. WOLOSHYN IS THE DIGITAL EFFECTS SUPERVISOR AT RAINMAKER FOR BOTH *STARGATE ATLANTIS* AND *STARGATE SG-1*. HE HAS BEEN WITH THE STARGATE FRANCHISES FROM THE VERY BEGINNING, STARTING WITH THE PILOT OF *STARGATE SG-1* BACK IN 1997 AND HAS BEEN HONORED WITH FOUR EMMY NOMINATIONS FOR HIS OUTSTANDING WORK AS LEAD DIGITAL COMPOSITING ARTIST ON THE SHOW. BRUCE'S OTHER VISUAL EFFECTS TELEVISION CREDITS INCLUDE GENE RODDENBERRY'S *ANDROMEDA, HIGHLANDER,* JAMES CAMERON'S *DARK ANGEL, MILLENNIUM,* AND THE PILOT TO *SMALLVILLE*. HIS FEATURE FILM CREDITS INCLUDE SUCH TITLES AS *ANTITRUST, MISSION TO MARS,* AND *3000 MILES TO GRACELAND*. BRUCE AND HIS VERY SUPPORTIVE WIFE RAMONA MAKE THEIR HOME BY THE SEA IN SUNNY SOUTH SURREY, BRITISH COLUMBIA, WITH THEIR TWO WONDERFUL SONS.

CULES: THE LEGENDARY JOURNEYS HYPERNAUTS INSIDE FORMULA 1 INVADER ZIM JAG JUSTICE LEAGUE
EN SISCO KINGDOM HOSPITAL LIVINGSTONE LOST IN SPACE MAD ABOUT YOU MAX STEEL MYSTIC KNIGHTS OF TIR NA NOG
CAR RACERS NAVY NCIS NIGHTMAN PARIS UTOPIES PECOLA THE PJS POWER PUFF GIRLS POWER RANGERS: LOST GALAX
ERS ROBOCOP: THE SERIES ROUGHNECKS: STARSHIP TROOPERS SEAQUEST DSV SEVEN DAYS SILVER SURFER: ANIMATE
ES SLIDERS SMALLVILLE SPACE: ABOVE AND BEYOND STARGATE ATLANTIS STARGATE SG-1 STAR TREK: DEEP SPACE NIN
R TREK: ENTERPRISE STAR TREK: VOYAGER TIMECOP THE TONIGHT SHOW THREAT MATRIX UNSOLVED MYSTERIES VIPER TH
OR WEIRD SCIENCE THE X-FILES XENA: WARRIOR PRINCESS XYBER 9: NEW DAWN YOU WISH! 12 DAYS OF TERROR 20,00
GUES UNDER THE SEA AFTERSHOCK: EARTHQUAKE IN NEW YORK AMERICAN MUSIC AWARDS BALL AND CHAIN BATTLESTA
ACTICA BBC: BAFTA AWARDS BLOOD: THE LAST VAMPIRE A BRIGHT SHINING LIE BUGS CATEGORY 6: DAY OF DESTRUCTIO
DREN OF DUNE DINOTOPIA DNA DUNE THE ELIAN GONZALES STORY FROM THE EARTH TO THE MOON GOLDEN GLOBE AWARD
VIOUR HAPPY BIRTHDAY HORIZON "T-REX: WARRIOR OR WIMP?" THE HUNLEY I SAW MOMMY KISSING SANTA CLAUS INVASIO
RICA IT CAME FROM OUTER SPACE II JAMES BOND: A BAFTA TRIBUTE LES GÉNIES DE LA MER LOST DINOSAURS OF EGYPT 300
S TO GRACELAND BATTLEFIELD EARTH CHARLIE'S ANGELS THE AVIATOR BLADE TRINITY THE CELL DRACULA 2000 DUNGEON
DRAGONS HOW THE GRINCH STOLE CHRISTMAS EXORCIST: THE BEGINNING FINAL DESTINATION GLADIATOR LITTLE NICK
SION TO MARS O BROTHER, WHERE ART THOU? THE DAY AFTER TOMORROW PITCH BLACK RED PLANET RULES OF ENGAGEMEN
RY MOVIE THE SIXTH DAY SUPERNOVA X-MEN FLIGHT OF THE PHOENIX TOY STORY 2 GARFIELD BATS THE ASTRONAUT'S WIF
P BLUE SEA DEEP END OF THE OCEAN HARRY POTTER AND THE PRISONER OF AZKABAN DOGMA FROM DUSK TILL DAWN 2 M
ORITE MARTIAN THE FIFTH ELEMENT SKY CAPTAIN AND THE WORLD OF TOMORROW ARMAGEDDON THE THOMAS CROWN AFFA
WORLD IS NOT ENOUGH LOST IN SPACE HELLBOY BLADE THE AVENGERS SPIDER-MAN 2 THE BIG HIT BLUES BROTHERS 200
IRMAN OF THE BOARD DEEP IMPACT I, ROBOT OCTOPUS CONTACT PHANTOMS VAN HELSING SHAKESPEARE IN LOVE THE SIEG
E OF THE MUMMY JURASSIC PARK TORQUE RED CORNER CONSPIRACY THEORY FINDING NEMO ANASTASIA DEVIL'S ADVOCAT
BBER FREE WILLY 3: THE RESCUE DAREDEVIL SPAWN MEN IN BLACK THE JACKAL GOTHIKA MORTAL KOMBAT II: ANNIHILATIO
RAGE UNDER FIRE GALAXIS THUNDERBIRDS GHOST IN THE SHELL 2: INNOCENCE IMMORTELLE (AD VITAM) INNOCENCE SPRUN
HMASTER LORD OF ILLUSIONS LARA CROFT TOMB RAIDER 2: CRADLE OF LIFE STAR KID (AKA THE WARRIOR OF WAVERL
EET) NIXON GOLDENEYE FREDDY VS JASON TITANIC WATERWORLD FREEJACK S.W.A.T. STAR TREK VI: THE UNDISCOVERE
NTRY THE ITALIAN JOB THE MANCHURIAN CANDIDATE THE LEAGUE OF EXTRAORDINARY GENTLEMENL'APETTA GIULIA E L
NORA VITA LEGALLY BLONDE 2: RED, WHITE, AND BLONDE AGENT CODY BANKS THE LAST SAMURAI THE ADVENTURES OF JIMM
TRON, BOY GENIUS THE PASSION OF THE CHRIST SOMETHING BORROWED LORD OF THE RINGS: THE RETURN OF THE KING FINA
TINATION 2 DUPLEX LOONEY TUNES: BACK IN ACTION HOLES THE HOT CHICK THE MATRIX RELOADED THE TRIPLETS O
LVILLE THE PERFECT SCORE X2: X-MEN UNITED RULES OF ENGAGEMENT THE MATRIX REVOLUTIONS DUNGEONS AND DRAGON
PLANET PIRATES OF THE CARIBBEAN: THE CURSE OF THE BLACK PEARL THE AVIATOR BLADE TRINITY THE DAY AFTE
ORROW EXORCIST: THE BEGINNING FLIGHT OF THE PHOENIX GARFIELD GHOST IN THE SHELL 2: INNOCENCE HARRY POTTE
THE PRISONER OF AZKABAN HELLBOY IMMORTELLE (AD VITAM) INNOCENCE I, ROBOT THE MANCHURIAN CANDIDATE TH
SION OF THE CHRIST THE PERFECT SCORE SKY CAPTAIN AND THE WORLD OF TOMORROW SOMETHING BORROWED SPIDER-MA
UNDERBIRDS TORQUE VAN HELSING AGENT CODY BANKS DAREDEVIL DUPLEX FINAL DESTINATION 2 FINDING NEMO FREDDY V
ON GOTHIKA HOLES THE HOT CHICK THE ITALIAN JOB L'APETTA GIULIA E LA SIGNORA VITA LARA CROFT TOMB RAIDER
DLE OF LIFE THE LAST SAMURAI THE LEAGUE OF EXTRAORDINARY GENTLEMEN LEGALLY BLONDE 2: RED, WHITE, AND BLONE
QUE THE TRIPLETS OF BELLVILLE LOONEY TUNES: BACK IN ACTION THE MATRIX RELOADED THE MATRIX REVOLUTIONS S.W.A.
X-MEN UNITED ACCESS HOLLYWOOD THE ADVENTURES OF JIMMY NEUTRON, BOY GENIUS ALIEN HUNTER ALLY MCBEA
YLON 5 BAYWATCH THE BILL CSI MIAMI LORD OF THE RINGS: THE RETURN OF THE KING BATTLESTAR GALACTICA - THE SERIE
DS OF PREY BLOODY BRITAIN SCARLET PIRATES OF THE CARIBBEAN: THE CURSE OF THE BLACK PEARL BUFFY THE VAMPIR
YER BUTT-UGLY MARTIANS CAPTAIN SCARLET CENTURY CITY CHARMED CRUSADE CSI NEW YORK DAN DARE, PILOT OF TH
URE DARK ANGEL DARK SKIES DAWSON'S CREEK DEAD LIKE ME DEXTER'S LAB ENTERPRISEFAMILY GUY FIREFLY GARY AN
E HAUNTED HERCULES: THE LEGENDARY JOURNEYS HYPERNAUTS INSIDE FORMULA 1 INVADER ZIM JAG JUSTICE LEAGUE JIMM
TRON KAREN SISCO KINGDOM HOSPITAL LIVINGSTONE LOST IN SPACE MAD ABOUT YOU MAX STEEL MYSTIC KNIGHTS OF TIR N
NASCAR RACERS NAVY NCIS NIGHTMAN PARIS UTOPIES PECOLA THE PJS POWER PUFF GIRLS POWER RANGERS: LOST GALAX
VERS ROBOCOP: THE SERIES ROUGHNECKS: STARSHIP TROOPERS SEAQUEST DSV SEVEN DAYS SILVER SURFER: ANIMATE
IES SLIDERS SMALLVILLE SPACE: ABOVE AND BEYOND STARGATE ATLANTIS STARGATE SG-1 STAR TREK: DEEP SPACE NI
R TREK: ENTERPRISE STAR TREK: VOYAGER TIMECOP THE TONIGHT SHOW THREAT MATRIX UNSOLVED MYSTERIES VIPER TI
TOR WEIRD SCIENCE THE X-FILES XENA: WARRIOR PRINCESS XYBER 9: NEW DAWN YOU WISH! 12 DAYS OF TERROR 20,0
GUES UNDER THE SEA AFTERSHOCK: EARTHQUAKE IN NEW YORK AMERICAN MUSIC AWARDS BALL AND CHAIN BATTLESTA
ACTICA BBC: BAFTA AWARDS BLOOD: THE LAST VAMPIRE A BRIGHT SHINING LIE BUGS CATEGORY 6: DAY OF DESTRUCTIO
LDREN OF DUNE DINOTOPIA DNA DUNE THE ELIAN GONZALES STORY FROM THE EARTH TO THE MOON GOLDEN GLOBE AWAR
AVIOUR HAPPY BIRTHDAY HORIZON "T-REX: WARRIOR OR WIMP?" THE HUNLEY I SAW MOMMY KISSING SANTA CLAUS INVASIO
ERICA IT CAME FROM OUTER SPACE II JAMES BOND: A BAFTA TRIBUTE LES GÉNIES DE LA MER LOST DINOSAURS OF EGYP
INY MAX Q NASA: MARS SPIRIT LANDER NATIONAL LAMPOON'S MEN IN WHITE OLIVE, THE OTHER REINDEER THE PRESIDENT
: A LINE
QUICKSILVER HIGHWAY SANTA VS. THE SNOWMAN SECRETS OF A HOMICIDE: THE JFK ASSASSINATIC
R EMMA
EXCEPTION SNOW WHITE SUPERFIRE THE SIN EATER TAKEN TITANIC TOM CLANCY'S NETFORC
AGE OF
DAYS A WRINKLE IN TIME DEEP BLUE SEA DEEP END OF THE OCEAN FAVORITE MARTIAN TI
GEONS A
E GRINCH STOLE CHRISTMAS THE DAY AFTER TOMORROW HELLBOY BLADE SPIDER-MAN 2 TH
LIGHTWAVE 3D
model animate render

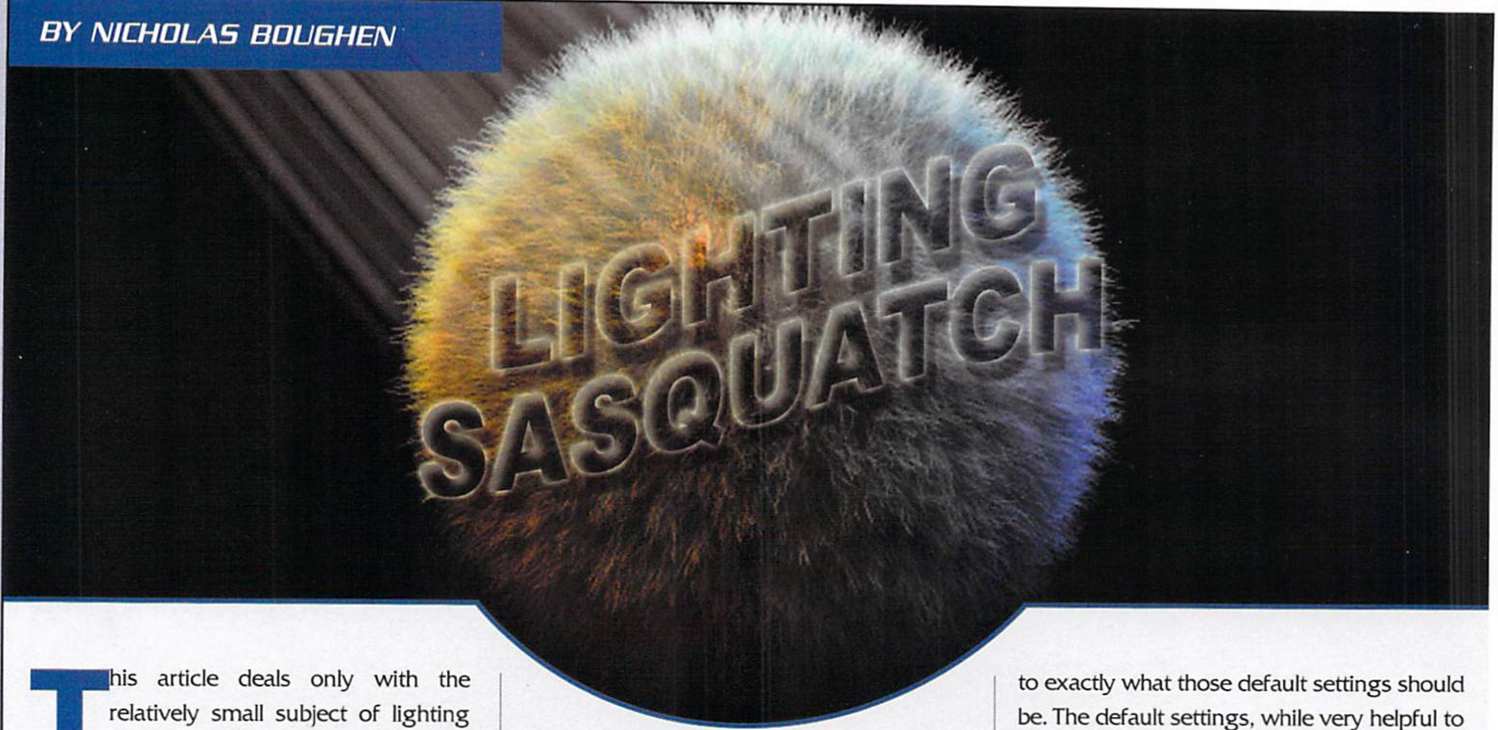
LightWave 3D®: Behind the scenes more often than not.

For more information about the Emmy Award Winning LightWave®
call 1 800 843 8934 or visit www.lightwave3d.com/hdri3d

© 2005. All rights reserved. LightWave and LightWave 3D are registered trademarks of NewTek. Emmy is a registered trademark of the Academy of Television Arts and Sciences. NewTek is the proud recipient of the 2003 Primetime Emmy Award for Engineering for LightWave 3D.

BY NICHOLAS BOUGHEN

LIGHTING SASQUATCH



This article deals only with the relatively small subject of lighting and shading in Sasquatch. This means most of the tools in Sasquatch won't be dealt with. This is just fair warning—if you're having trouble with fur density, length, kink, frizz, curl, combing, or any of the other zillion parameters, you'll have to find the answers elsewhere.

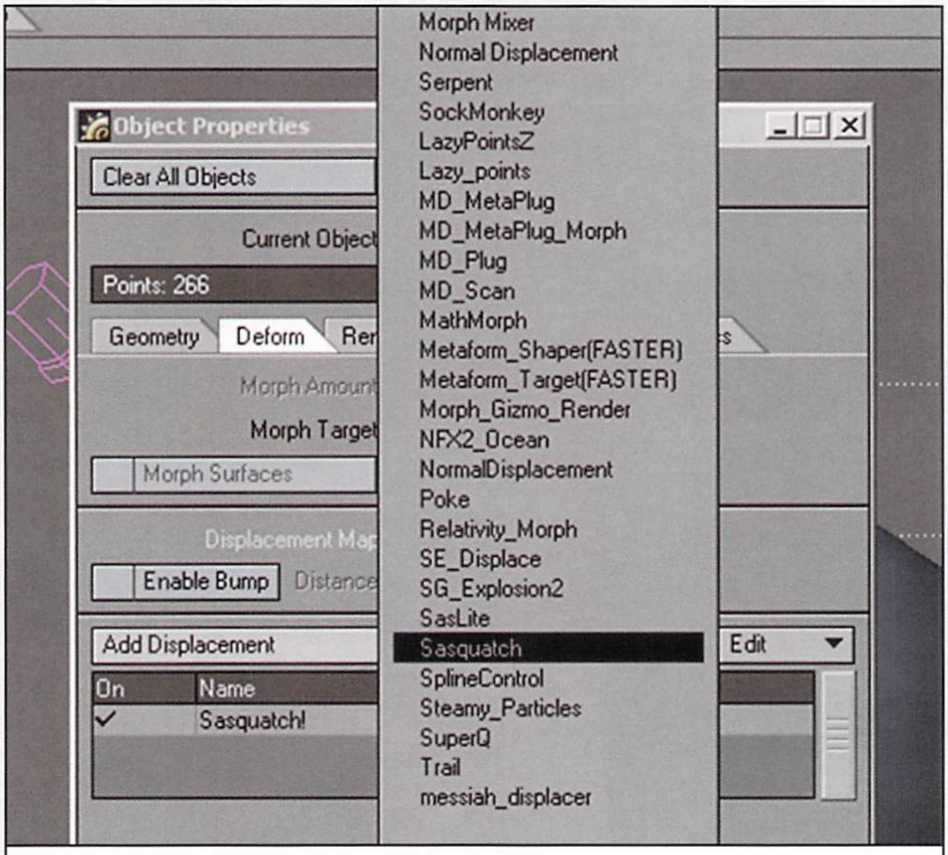
I commonly read and hear of LightWave artists having difficulty getting Sasquatch to look good and behave predictably under lighting. We all know what happens when we turn on a light, aim it at a model, and set the color and intensity, but once we add Sasquatch fur, the results can be unexpected if we aren't careful.

The main reason things don't always work out as expected is twofold.

First, Mr. Worley, author of Sasquatch, painstakingly provided us with many tools for shading the fur and hair fibers. This awesome power, however, is a double-edged sword. It means we have great control over how the fur fibers look, but it also means we need to know exactly how to drive all the controls. We also need to know how each tool interacts with its neighbors. It's like the difference between a 1930 biplane's control panel and that of a jumbo jet. You can do a lot more with a jumbo jet—you can definitely fly higher and go farther and faster, but if you're not intimately familiar with the controls, you're probably going to crash, if in fact you manage to get off the ground at all. Sasquatch is much like this. It provides you with vast power, but you'll do your best if you learn exactly how the switches and dials really work.

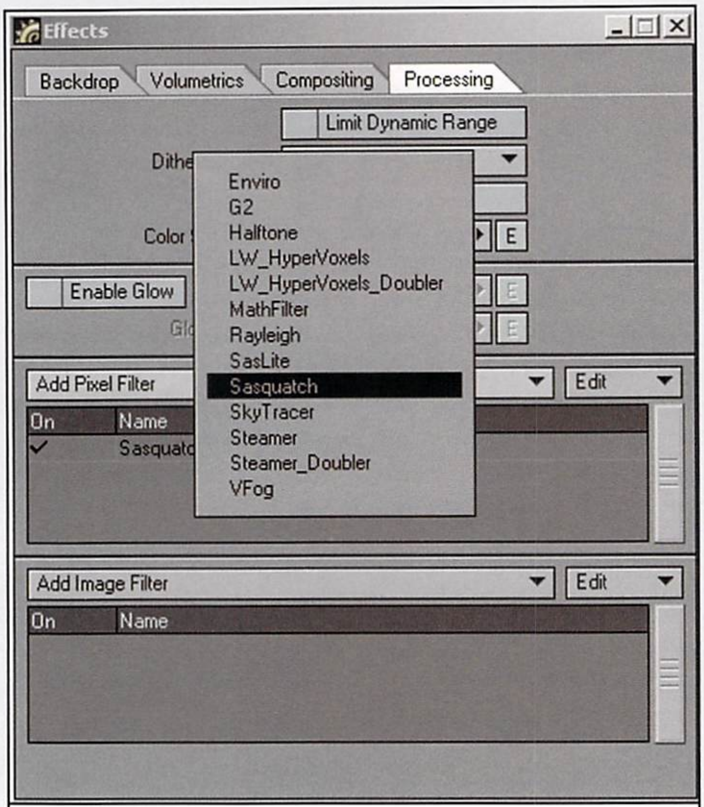
Second, Mr. Worley has attempted to make our jobs easier by providing default settings. Rumor has it there was great debate during the beta testing phase of Sasquatch as to exactly what those default settings should be. The default settings, while very helpful to get started, can be detrimental once you get into more advanced lighting.

Note: If you are not already familiar with Sasquatch, two plug-ins are required to get it to render. First, make sure you have the object selected to which you want to add fur or fibers. Now, on the object deformation tab, add the Sasquatch plug-in.


Object Properties
Clear All Objects
Current Object
Points: 266
Geometry
Deform
Morph Amount
Morph Target
Morph Surfaces
Displacement Map
Enable Bump
Distance
Add Displacement
Edit
On
Name
Sasquatch!
Morph Mixer
Normal Displacement
Serpent
SockMonkey
LazyPointsZ
Lazy_points
MD_MetaPlug
MD_MetaPlug_Morph
MD_Plug
MD_Scan
MathMorph
Metaform_Shaper(FASTER)
Metaform_Target(FASTER)
Morph_Gizmo_Render
NFX2_Ocean
NormalDisplacement
Poke
Relativity_Morph
SE_Displace
SG_Explosion2
SasLite
Sasquatch
SplineControl
Steamy_Particles
SuperQ
Trail
messiah_displacer

ADD THE SASQUATCH DISPLACEMENT FROM THE OBJECTS>PROPERTIES>ADD DISPLACEMENT DROPDOWN

Then, in the Image Processing panel, add Sasquatch in the Pixel Filter dropdown.

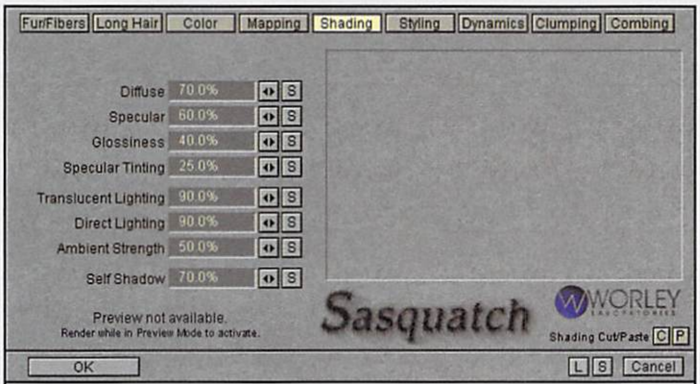

Effects
Backdrop
Volumetrics
Compositing
Processing
Limit Dynamic Range
Dithe
Color
E
Enable Glow
Gl
E
E
Add Pixel Filter
Edit
On
Name
Sasquatc
Enviro
G2
Halftone
LW_HyperVoxels
LW_HyperVoxels_Doubler
MathFilter
Rayleigh
SasLite
Sasquatch
SkyTracer
Steamer
Steamer_Doubler
VFog
Add Image Filter
Edit
On
Name

ADD THE SASQUATCH PIXEL FILTER IN THE EFFECTS>PROCESSING>ADD PIXEL FILTER DROPDOWN

This article is split into two sections. "The Essentials" covers things you really need to know to get good looking lighting and shading out of Sasquatch. "The Extras" covers additional tools that can help you further. If you're a serious fur and hair artist, you'll want to know everything here. If you're in a hurry, just read "The Essentials" and cover the rest when you have time.

THE ESSENTIALS

So we're dealing only with lighting and shading in Sasquatch today. This takes place in two panels. One is the Deformation plug-in *Shading* panel.

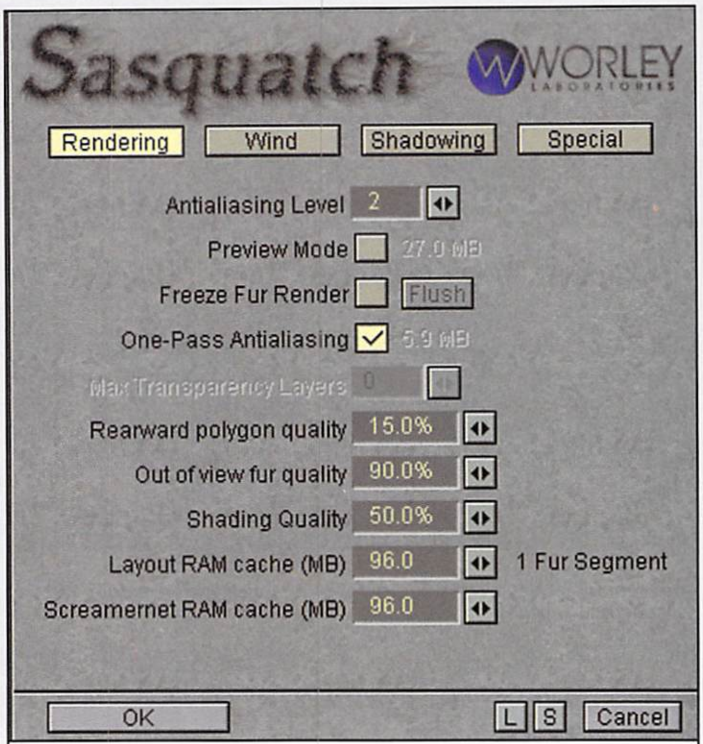

Fur/Fibers
Long Hair
Color
Mapping
Shading
Styling
Dynamics
Clumping
Combing
S
Sasquatch
WORLEY LABORATORIES
Shading Cut/Paste
C
P
OK
L
S
Cancel

Diffuse 70.0%
Specular 60.0%
Glossiness 40.0%
Specular Tinting 25.0%
Translucent Lighting 90.0%
Direct Lighting 90.0%
Ambient Strength 50.0%
Self Shadow 70.0%

Preview not available.
Render while in Preview Mode to activate.

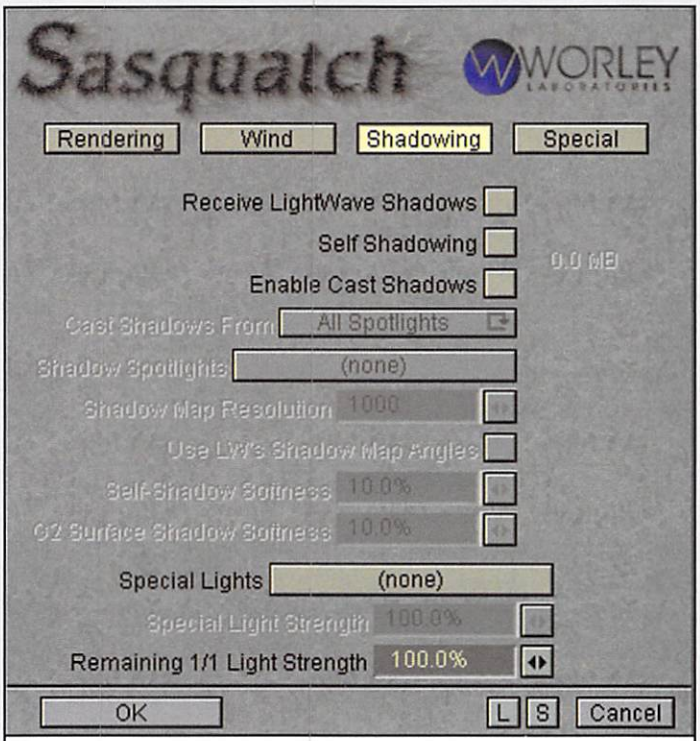
THE SASQUATCH SHADING TAB

The other is on the *Rendering* and *Shadowing* panels of the pixel filter.


Sasquatch
WORLEY LABORATORIES
Rendering
Wind
Shadowing
Special
Flush
OK
L
S
Cancel

Antialiasing Level 2
Preview Mode ☐ 27.0 MB
Freeze Fur Render ☐
One-Pass Antialiasing ☑ 5.9 MB
Max Transparency Layers 0
Rearward polygon quality 15.0%
Out of view fur quality 90.0%
Shading Quality 50.0%
Layout RAM cache (MB) 96.0 1 Fur Segment
Screamernet RAM cache (MB) 96.0

THE SASQUATCH PIXEL FILTER RENDERING TAB

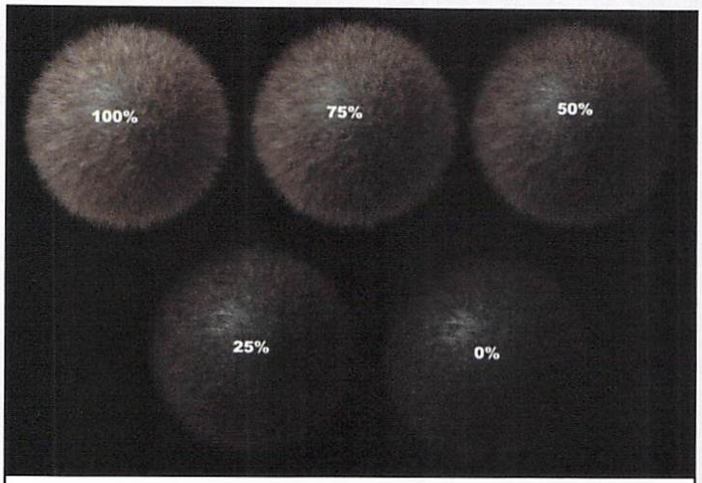

Sasquatch
WORLEY LABORATORIES
Rendering
Wind
Shadowing
Special
OK
L
S
Cancel

Receive LightWave Shadows ☐
Self Shadowing ☐ 0.0 MB
Enable Cast Shadows ☐
Cast Shadows From All Spotlights
Shadow Spotlights (none)
Shadow Map Resolution 1000
Use LW's Shadow Map Angles ☐
Self-Shadow Softness 10.0%
G2 Surface Shadow Softness 10.0%
Special Lights (none)
Special Light Strength 100.0%
Remaining 1/1 Light Strength 100.0%

THE SASQUATCH PIXEL FILTER SHADOWING TAB

Let's deal with the Sasquatch settings one at a time. To begin with, on the Deformation plug-in's Shading Panel, we have our old familiar Diffuse, Specularity, and Glossiness.

Just as with any LightWave texture, the **Diffuse** value determines how much illumination a texture (or in this case, a fiber) receives from a light. For example, if a light is set at 100% and the Diffuse value is set at 50%, then the fiber will receive (or, technically, reflect) 50% of the light.



FUR DIFFUSE VALUES CHANGE THE BRIGHTNESS OF THE FUR FIBERS

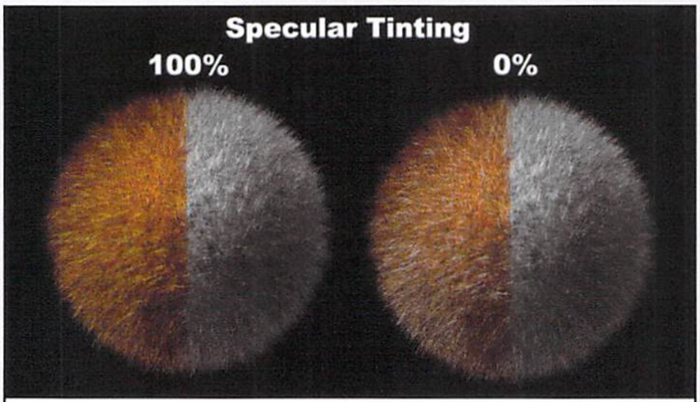
All fur settings are default, except Diffuse, as noted in the illustration. Note that the ball texture does not change when the Diffuse value of the fur is changed, as fur settings are completely independent of the geometry's texture settings.

The **Specularity** parameter controls the brightness of the specular highlight, and the **Glossiness** parameter controls how shiny (or small) the specular highlight is. Materials that are very smooth will have a much smaller specular highlight.

NOTE: It is important to note that, while Specularity and Glossiness controls work the same way they do in LightWave, the actual specular highlights are not the same. Because each hair fiber is essentially a long tube, the specular highlights are stretched along the length of the fiber.

This is where we leave the familiar world of LightWave parameters and delve into unique Sasquatch settings.

Normally, the specular highlight will be the color of the light that is creating the highlight. But **Specular Tinting** is a great tool that allows you to let some of the fiber color into that specular highlight. If, for example, you have a calico cat, then some of the orange will bleed into specular highlights on that part of the fur, while specular highlights on the white fur will remain white. Specular Tinting will tend to make the specular highlights subtler and less visible, since they blend in with the color of the material instead of being a white highlight on top of it, but the highlights will also behave more realistically.



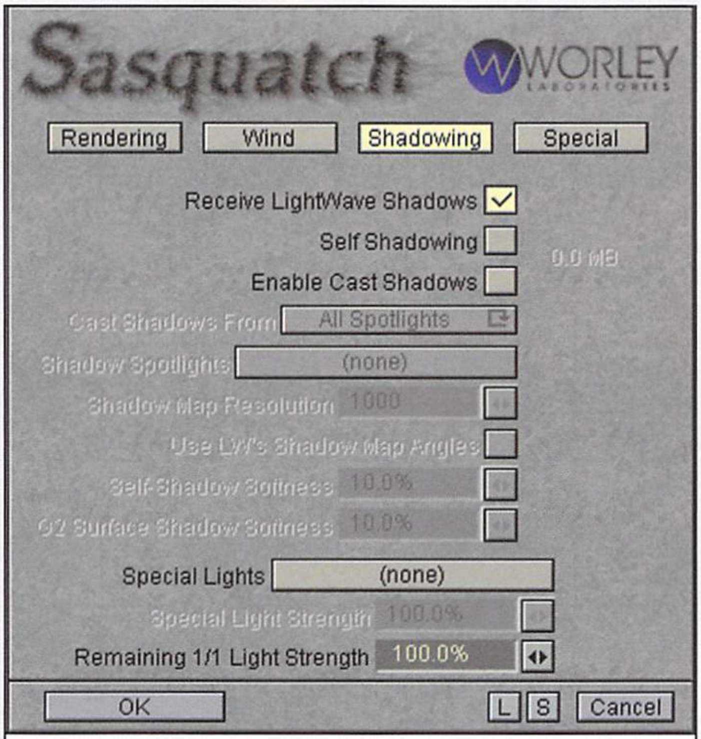
SPECULAR TINTING ADDS FUR COLOR TO THE SPECULAR HIGHLIGHT

The settings on these two images are identical except for specular tinting. With Specular Tinting at 100%, the specular highlight strongly takes on the color of the fiber. This actually makes the specular highlight difficult to see. With Specular Tinting set to 0%, none of the fiber color comes through in the highlight, and the highlight color is determined entirely by the color of the light source. The best looking and most visible specular highlight will probably be somewhere in between 0% and 100%, where the highlight is easily visible, but also takes on some of the color from the fiber.

Translucent Lighting and **Direct Lighting** fall into the category of "Beyond Here There Be Dragons." It's very easy to get in trouble by using these two values to set up your lighting. If you find you have to move the object or the light, suddenly the fur lighting behaves differently and you have to relight the shot.

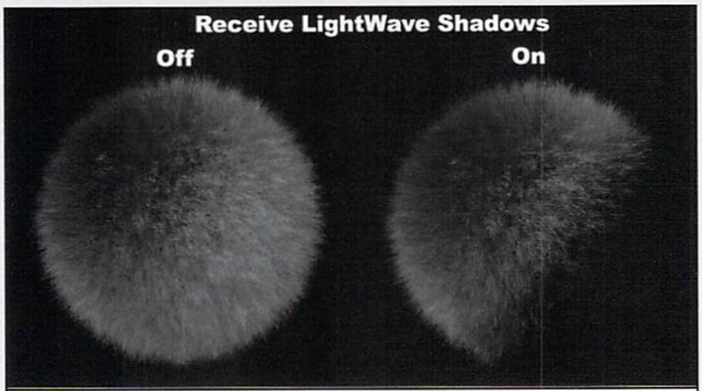
I strongly recommend beginners to follow these three steps:

1 In your pixel filter, make sure you have "Receive LightWave Shadows" turned on. This will ensure that light will not pass through your geometry and light the fur on the far side (this is the single highest cause of consternation among new Sasquatchers).



Receive LightWave Shadows ☑
Self Shadowing ☐
Enable Cast Shadows ☐
0.0 MB
Cast Shadows From: All Spotlights
Shadow Spotlights: (none)
Shadow Map Resolution: 1000
Use LW's Shadow Map Angles ☐
Self-Shadow Softness: 10.0%
O2 Surface Shadow Softness: 10.0%
Special Lights: (none)
Special Light Strength: 100.0%
Remaining 1/1 Light Strength: 100.0%

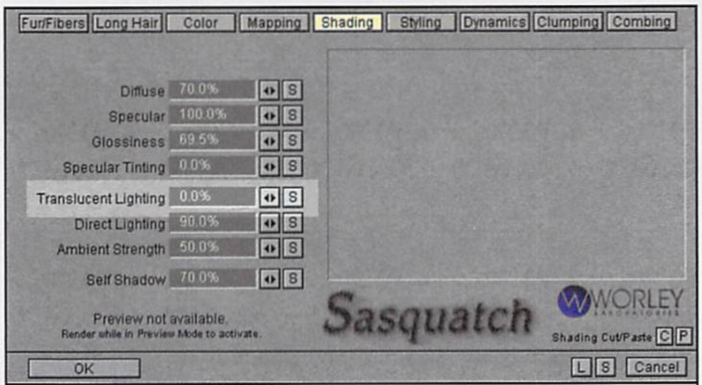
"RECEIVE LIGHTWAVE SHADOWS" IS TURNED ON



WHEN THE FUR RECEIVES LIGHTWAVE SHADOWS, ANY LIGHTWAVE GEOMETRY BLOCKS THE LIGHT RAYS

Both these images have a spotlight source above and to the left, except the left image appears to have an additional light source below and to the right, because otherwise, the bottom right area of the fur would be in shadow. However, Sasquatch is not receiving LightWave's shadows, so Sasquatch doesn't cut out the light that would normally be stopped by the geometry of the ball. In the right image, Sasquatch "sees" that the geometry is casting a shadow, so the bottom-right part of the ball is in shadow, just as it should be.

2 If you haven't turned on "Receive LightWave Shadows," then start with a Translucent Lighting value of 0%.



Diffuse 70.0%
Specular 100.0%
Glossiness 69.5%
Specular Tinting 0.0%
Translucent Lighting 0.0%
Direct Lighting 90.0%
Ambient Strength 50.0%
Self Shadow 70.0%

Preview not available.
Render while in Preview Mode to activate.

TRANSLUCENT LIGHTING SET TO 0% ON THE SHADING TAB

This will ensure that your initial lighting is working just as you'd expect it to. You'll see light on the correct side, just as it should be. Note: If you're already using "Receive LightWave Shadows," then you'll notice that Translucent Lighting tends to brighten and soften the look of the fibers. This is a powerful and subtle effect, making the fibers appear translucent, just as they often do in real life.

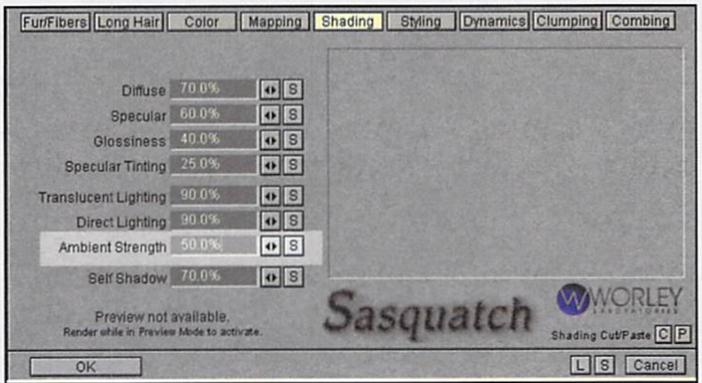
3 Always use spotlights with Shadow Maps if you can. Sasquatch calculates its own shadow maps. This will save you huge render time. Don't use ray-traced shadows if you can avoid it.

TRANSLUCENT LIGHTING VS. DIRECT LIGHTING

Direct Lighting works much the same as any lights in LightWave work on any geometry texture. Light is shone directly onto a texture and the texture is illuminated by it based on lighting intensity, angle, color, etc. With Translucent Lighting, the fiber seems to be given an almost luminous quality. The huge advantage to this is that, since fibers are thin lines, Direct Lighting doesn't do a good job on fibers that are pointing directly at the light source, because there is so little surface area, but Translucent Lighting will illuminate the fiber no matter which direction it is facing (provided it's not in shadow).

I always start with only Direct Lighting, and then add Translucent Lighting to fill and enhance.

Next on the list is "**Ambient Strength**."



Diffuse 70.0%
Specular 60.0%
Glossiness 40.0%
Specular Tinting 25.0%
Translucent Lighting 90.0%
Direct Lighting 90.0%
Ambient Strength 50.0%
Self Shadow 70.0%

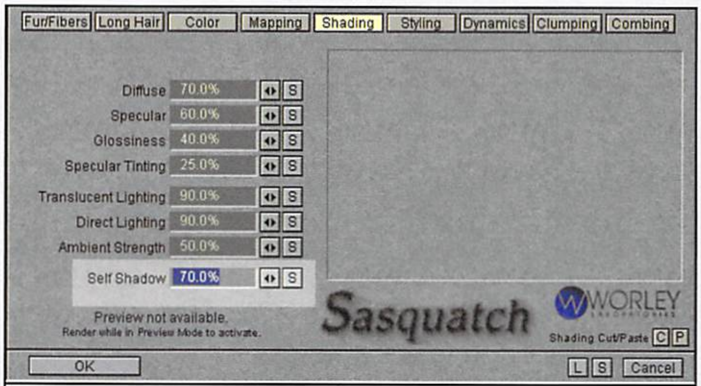
Preview not available.
Render while in Preview Mode to activate.

AMBIENT STRENGTH IS SET TO 50% ON THE SHADING TAB

This setting lets you determine how much of LightWave's ambient light will affect the fur fibers. If, for example, you have your Ambient at the LightWave default of 5%, but you don't want any ambient light on the fur, then you can set this to 0% to cut it out entirely. If, on the other hand, you want a high ambient light, without changing LightWave's settings, then you can set this value to 1000% to set the fur's ambient light intensity to 50%. LightWave's ambient parameter is multiplied by the Ambient Strength value to determine how much ambient light is received by the fur fibers.

Self Shadowing: This setting determines the shadow density of the shadows cast by the fur fibers onto other fur fibers

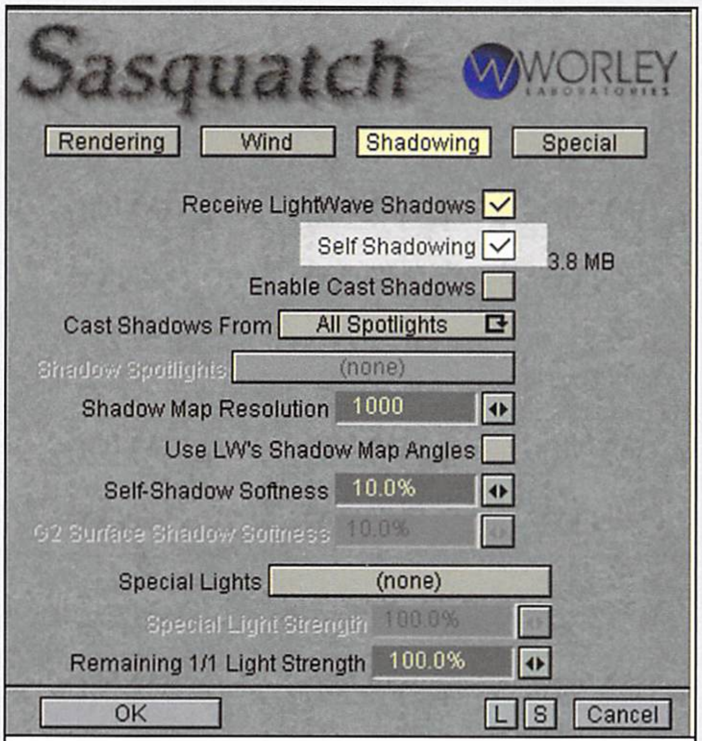
In order to use Self Shadowing, first you must turn on the Self Shadowing button in the Shadowing panel of the Sasquatch pixel filter.


Fur/Fibers
Long Hair
Color
Mapping
Shading
Styling
Dynamics
Clumping
Combing
Sasquatch
WORLEY
Shading Cut/Paste C P
OK
L S Cancel

Diffuse 70.0%
Specular 60.0%
Glossiness 40.0%
Specular Tinting 25.0%
Translucent Lighting 90.0%
Direct Lighting 90.0%
Ambient Strength 50.0%
Self Shadow 70.0%

Preview not available.
Render while in Preview Mode to activate.

THE "SELF SHADOW" SETTING LETS YOU DETERMINE THE OPACITY OF THE SHADOWS FROM ONE FUR FIBER TO ANOTHER. HIGHER VALUES MAKE DARKER SHADOWS

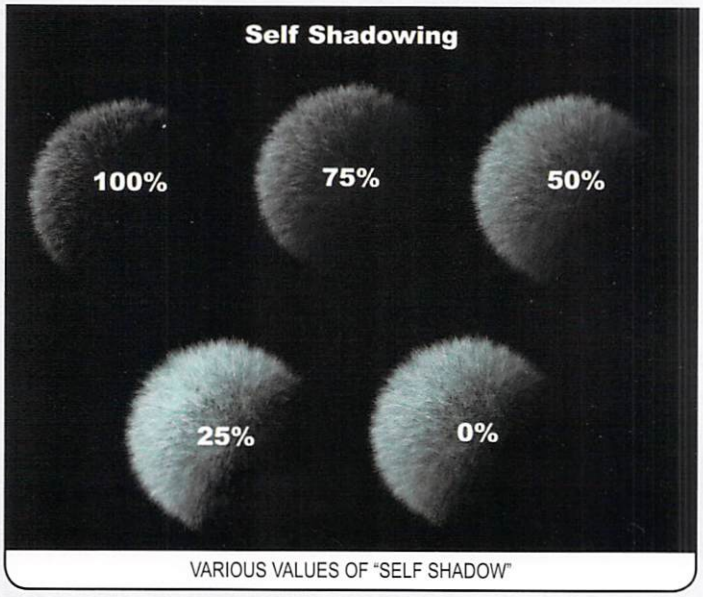

Sasquatch
WORLEY
Rendering
Wind
Shadowing
Special
OK
L S Cancel

Receive LightWave Shadows ☑
Self Shadowing ☑ 3.8 MB
Enable Cast Shadows ☐
Cast Shadows From All Spotlights
Shadow Spotlights (none)
Shadow Map Resolution 1000
Use LW's Shadow Map Angles ☐
Self-Shadow Softness 10.0%
G2 Surface Shadow Softness 10.0%
Special Lights (none)
Special Light Strength 100.0%
Remaining 1/1 Light Strength 100.0%

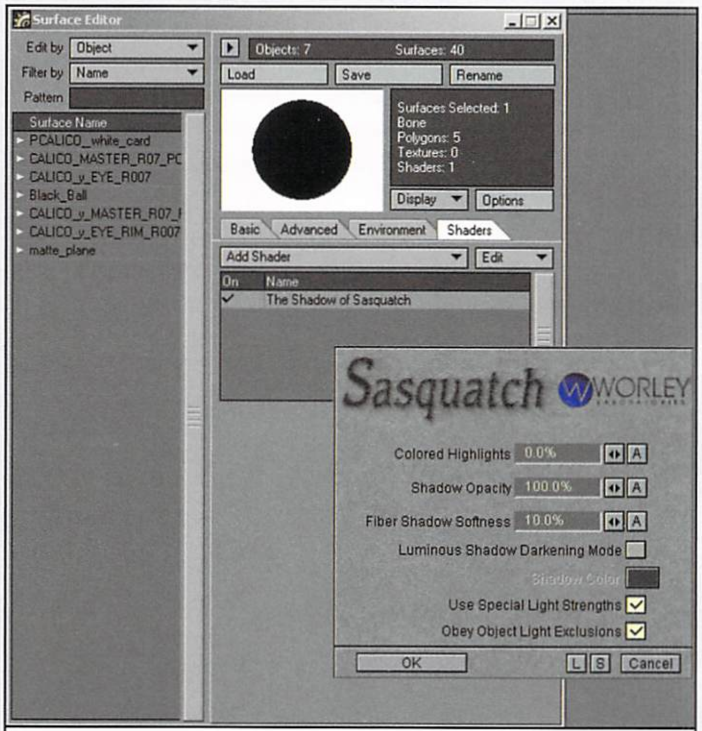
WITH "SELF SHADOWING" ON, FUR WILL CAST SHADOWS ONTO OTHER FUR FIBERS

The default setting is 70%. At the top right of this page are some different Self Shadowing settings.

If you wish to have your fur or hair fibers cast shadows onto other fibers or onto your geometry … say you have very bushy eyebrows over the eyes … then you'd want to turn on "Enable Cast Shadows" in the pixel filter. You also need to add a texture shader to your geometry called "The Shadow of Sasquatch."


Self Shadowing
100%
75%
50%
25%
0%

VARIOUS VALUES OF "SELF SHADOW"


Surface Editor
Load
Save
Rename
Display
Options
Basic
Advanced
Environment
Shaders
Add Shader
Edit
Sasquatch
WORLEY
OK
L S Cancel

Edit by Object
Filter by Name
Pattern

Objects: 7 Surfaces: 40

Surface Name
- PCALICO_white_card
- CALICO_MASTER_R07_PC
- CALICO_y_EYE_R007
- Black_Ball
- CALICO_y_MASTER_R07_I
- CALICO_y_EYE_RIM_R007
- matte_plane

Surfaces Selected: 1
Bone
Polygons: 5
Textures: 0
Shaders: 1

On	Name
✓	The Shadow of Sasquatch

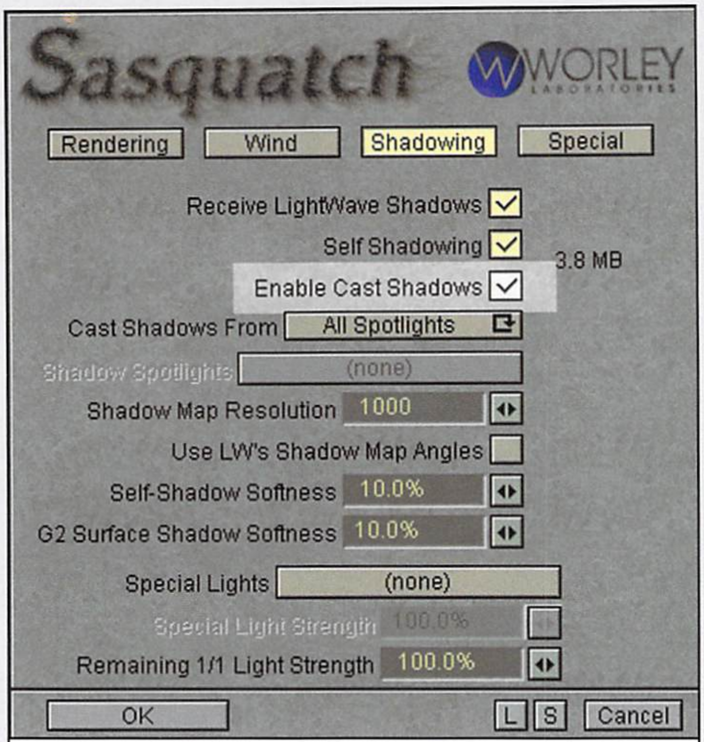
Colored Highlights 0.0%
Shadow Opacity 100.0%
Fiber Shadow Softness 10.0%
Luminous Shadow Darkening Mode ☐
Shadow Color
Use Special Light Strengths ☑
Obey Object Light Exclusions ☑

"THE SHADOW OF SASQUATCH" IS A SURFACE SHADER THAT ALLOWS GEOMETRY TO RECEIVE SHADOWS FROM SASQUATCH

The shader controls are fairly straightforward and are well covered in the Sas manual. Note: Remember that in order to cast fur shadows onto geometry, you must have *Enable Cast Shadows* turned on in the Sasquatch pixel filter.

Summing up, we want to ensure the following things:

- We are using spotlights with shadow maps
- We have *Receive LightWave Shadows* and *Self Shadowing* turned ON in the pixel filter.
- We start with a low Translucent Lighting setting, and then turn it up as needed.
- We are aware of how much ambient lighting we are receiving from LightWave.
- We are aware of how much "Self Shadowing" we are using on the fur fibers.

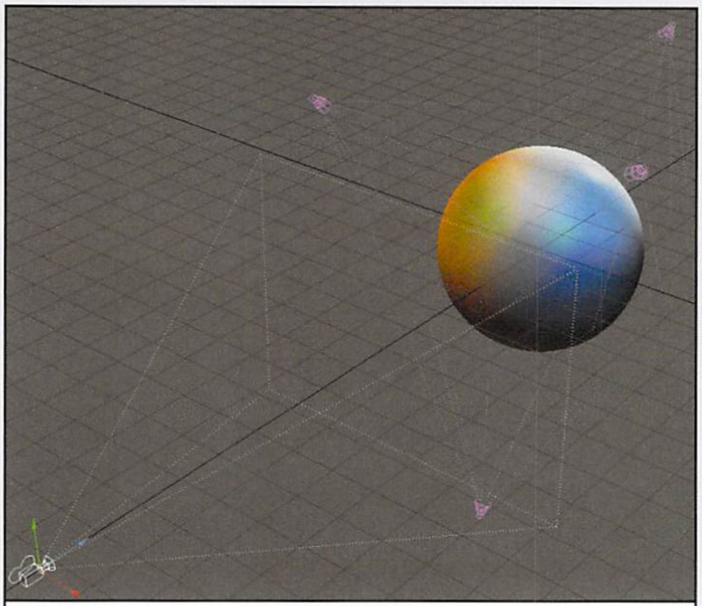

Sasquatch
WORLEY LABORATORIES
Rendering
Wind
Shadowing
Special
OK
L
S
Cancel

Receive LightWave Shadows ☑
Self Shadowing ☑
Enable Cast Shadows ☑
3.8 MB
Cast Shadows From: All Spotlights
Shadow Spotlights: (none)
Shadow Map Resolution: 1000
Use LW's Shadow Map Angles ☐
Self-Shadow Softness: 10.0%
G2 Surface Shadow Softness: 10.0%
Special Lights: (none)
Special Light Strength: 100.0%
Remaining 1/1 Light Strength: 100.0%

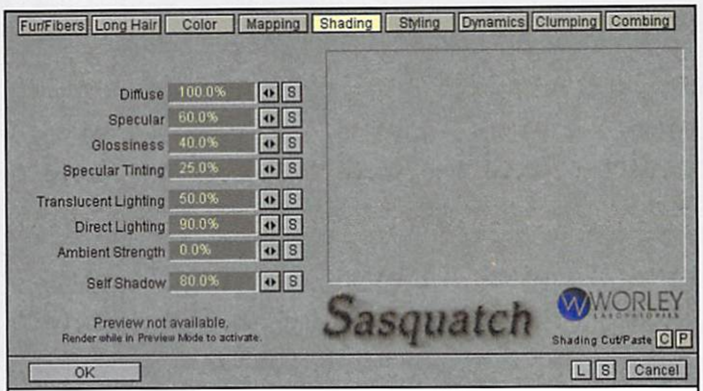
IN ORDER TO CAST FUR SHADOWS ONTO GEOMETRY, YOU MUST HAVE *ENABLE CAST SHADOWS* TURNED ON IN THE SASQUATCH PIXEL FILTER

Now that we have the lighting control basics down, we can set up a simple four-point rig, and see how our settings work.

Here are my final rig, settings, and render. You'll notice there's nothing special about the lighting rig at all. It's a very basic lighting rig, just as you'd use on any object. All that makes the fur look nice is a good understanding and skillful application of the shading settings.



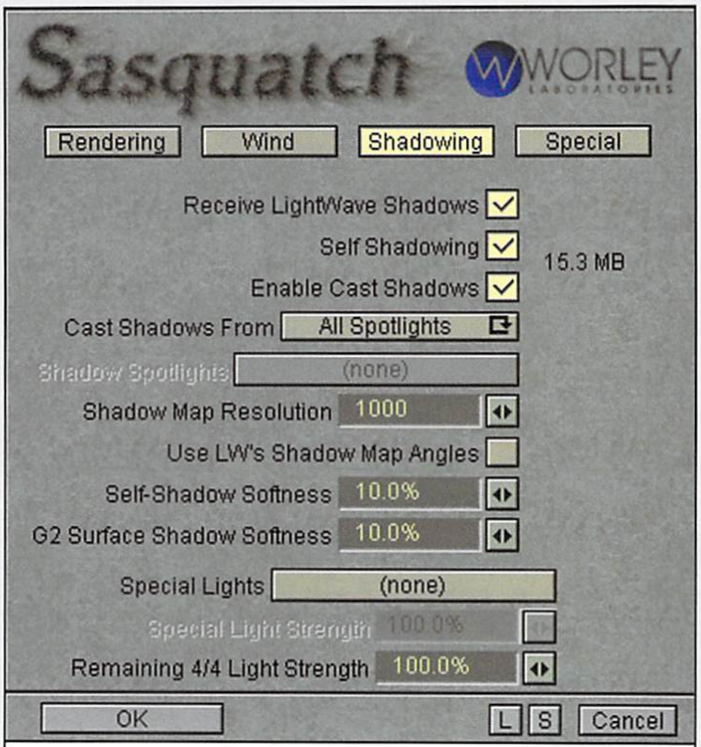
A SIMPLE FOUR-POINT LIGHT RIG


Fur/Fibers
Long Hair
Color
Mapping
Shading
Styling
Dynamics
Clumping
Combing
Sasquatch
WORLEY LABORATORIES
Shading Cut/Paste
C
P
OK
L
S
Cancel

Diffuse: 100.0%
Specular: 60.0%
Glossiness: 40.0%
Specular Tinting: 25.0%
Translucent Lighting: 50.0%
Direct Lighting: 90.0%
Ambient Strength: 0.0%
Self Shadow: 80.0%

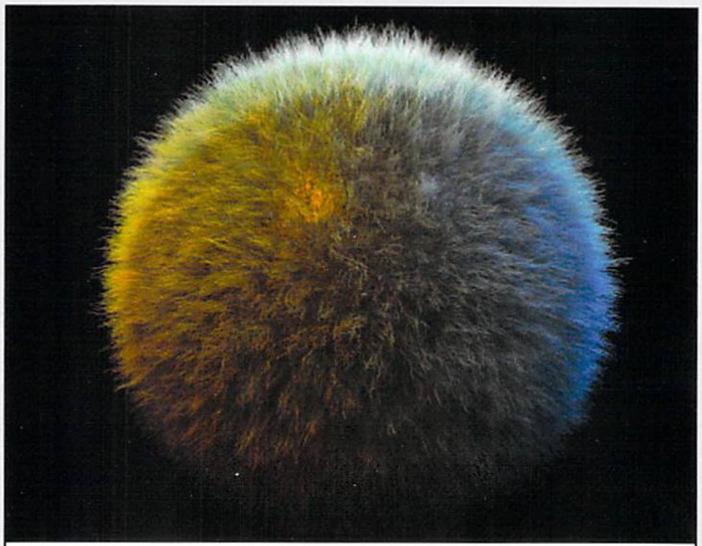
Preview not available.
Render while in Preview Mode to activate.

MY SASQUATCH SHADING SETTINGS


Sasquatch
WORLEY LABORATORIES
Rendering
Wind
Shadowing
Special
OK
L
S
Cancel

Receive LightWave Shadows ☑
Self Shadowing ☑
Enable Cast Shadows ☑
15.3 MB
Cast Shadows From: All Spotlights
Shadow Spotlights: (none)
Shadow Map Resolution: 1000
Use LW's Shadow Map Angles ☐
Self-Shadow Softness: 10.0%
G2 Surface Shadow Softness: 10.0%
Special Lights: (none)
Special Light Strength: 100.0%
Remaining 4/4 Light Strength: 100.0%

MY SASQUATCH PIXEL FILTER SETTINGS



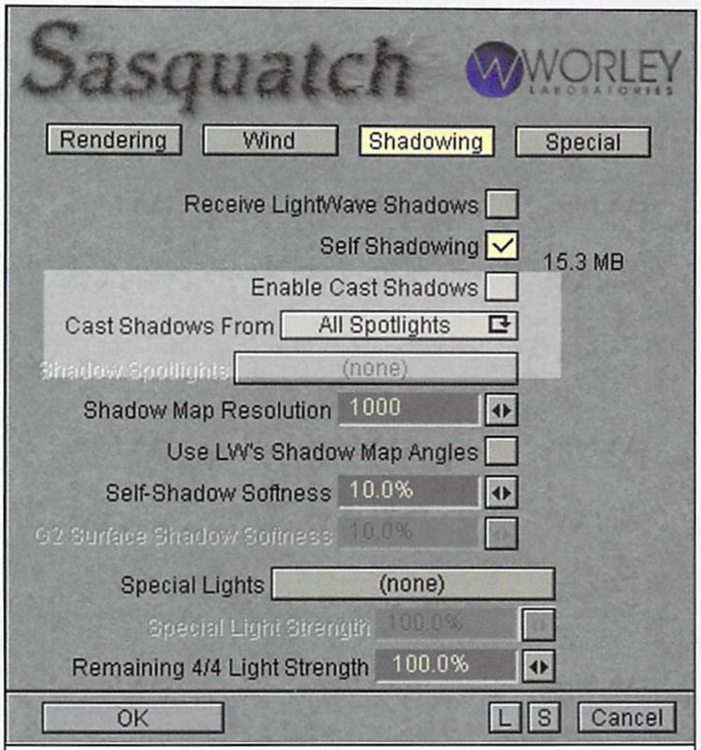
MY RENDER WITH THE ABOVE SETTINGS

THE EXTRAS

IN THE PIXEL FILTER, SHADOWING TAB

It's a small panel, but it holds serious control. We've already covered the first three toggles in the panel in the Essentials section, so let's move on.

CAST SHADOWS FROM



Receive LightWave Shadows ☐
Self Shadowing ☑
15.3 MB
Enable Cast Shadows ☐
Cast Shadows From: All Spotlights
Shadow Spotlights: (none)
Shadow Map Resolution: 1000
Use LW's Shadow Map Angles ☐
Self-Shadow Softness: 10.0%
G2 Surface Shadow Softness: 10.0%
Special Lights: (none)
Special Light Strength: 100.0%
Remaining 4/4 Light Strength: 100.0%

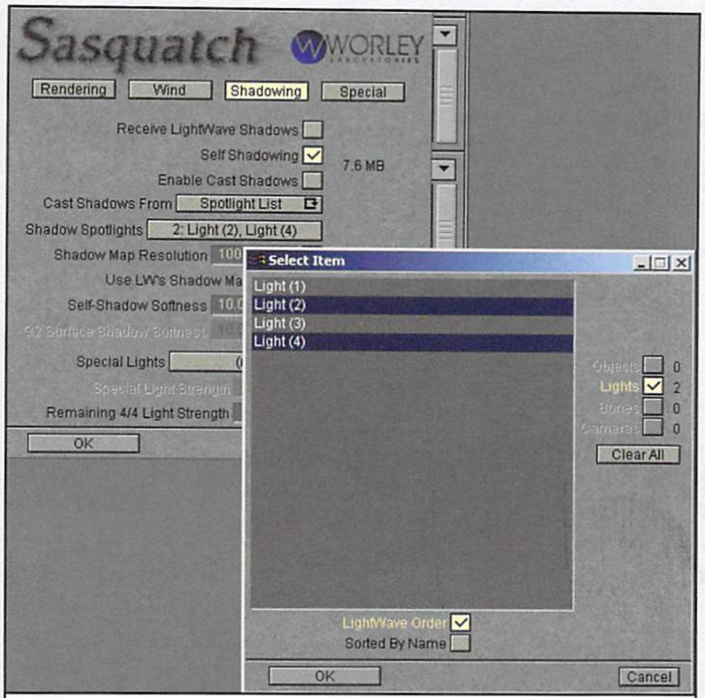
CHOOSING WHICH SPOTLIGHTS TO CAST SHADOWS FROM...

If you have *Self Shadowing* or *Enable Cast Shadows* turned on, you can force Sasquatch to select which lights it will cast shadows from. This defaults to *All Spotlights,* or you can switch to *LW Shadow Spots,* which means that shadows will be cast from all spotlights with shadow maps. You can also switch it to *Spotlight List,* which means you can pick only the spotlights you want from the list "Shadow Spotlights" is found just below this control. See image at right, top.

Sasquatch fibers will now receive shadows only from lights in the Shadow Spotlights list.

SHADOW MAP RESOLUTION

Sasquatch calculates its own shadow maps independently of LightWave's shadow maps, so naturally the shadow map resolution can be set independently as well. This is a good feature, since you'll often need a higher resolution for fur fibers than you will for geometry, since the fibers are so small and narrow. See image right, bottom.

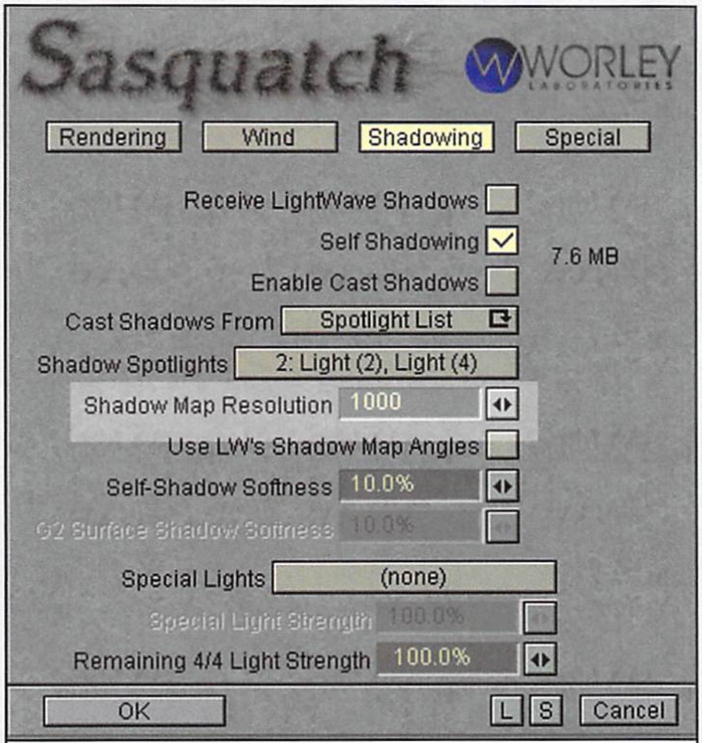


Receive LightWave Shadows ☐
Self Shadowing ☑
7.6 MB
Enable Cast Shadows ☐
Cast Shadows From: Spotlight List
Shadow Spotlights: 2: Light (2), Light (4)
Shadow Map Resolution: 100
Use LW's Shadow Ma
Self-Shadow Softness: 10.0
G2 Surface Shadow Softness: 10.0
Special Lights: (
Special Light Strength
Remaining 4/4 Light Strength

Light (1)
Light (2)
Light (3)
Light (4)

Objects ☐ 0
Lights ☑ 2
Bones ☐ 0
Cameras ☐ 0
LightWave Order ☑
Sorted By Name ☐

THIS IS THE LIST WHERE YOU SELECT WHICH SPOTLIGHTS TO CAST SHADOWS FROM



Receive LightWave Shadows ☐
Self Shadowing ☑
7.6 MB
Enable Cast Shadows ☐
Cast Shadows From: Spotlight List
Shadow Spotlights: 2: Light (2), Light (4)
Shadow Map Resolution: 1000
Use LW's Shadow Map Angles ☐
Self-Shadow Softness: 10.0%
G2 Surface Shadow Softness: 10.0%
Special Lights: (none)
Special Light Strength: 100.0%
Remaining 4/4 Light Strength: 100.0%

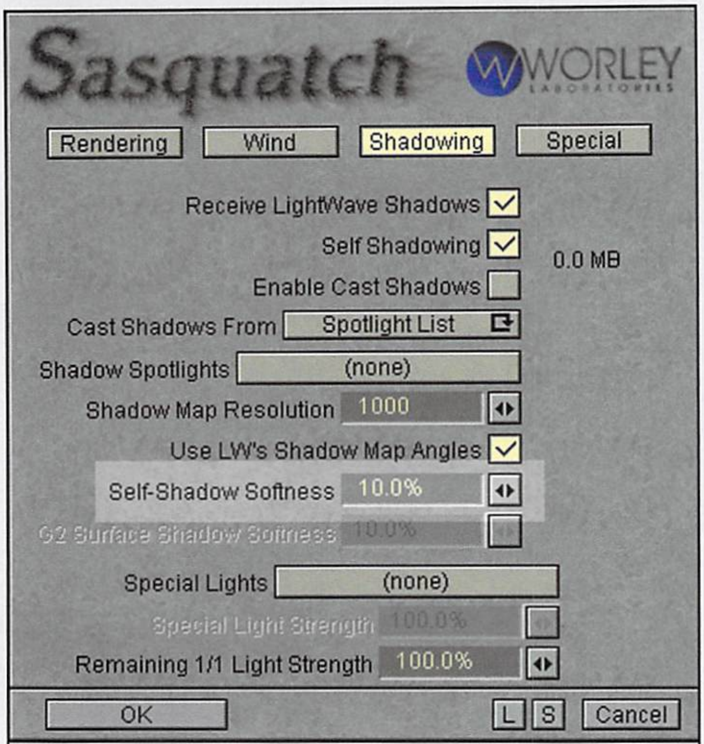
SASQUATCH SHADOW MAP RESOLUTION IS INDEPENDENT OF LIGHTWAVE'S SETTINGS

USE LW'S SHADOW MAP ANGLES

If you are using something other than the cone angle for your spotlight's shadow map, toggling this switch will cause Sasquatch to calculate its shadow maps at the correct angle.

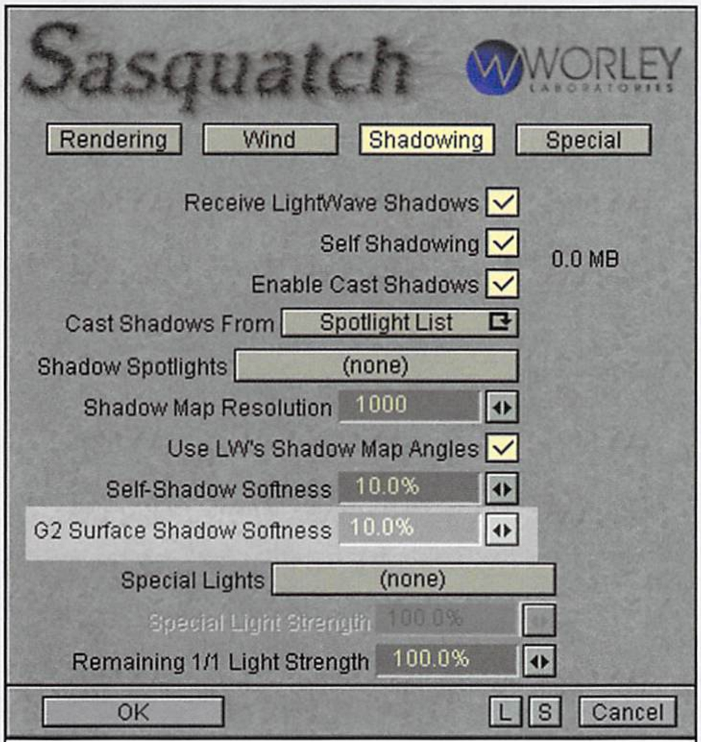
SELF-SHADOW SOFTNESS

Self-Shadow Softness is a great way to simulate shadow softness without having to go to the trouble of using expensive soft lighting solutions like area lights or even spotlight arrays. I've found myself using a setting of around 50% often.



Receive LightWave Shadows ☑
Self Shadowing ☑
Enable Cast Shadows ☐
0.0 MB
Cast Shadows From Spotlight List
Shadow Spotlights (none)
Shadow Map Resolution 1000
Use LW's Shadow Map Angles ☑
Self-Shadow Softness 10.0%
G2 Surface Shadow Softness 10.0%
Special Lights (none)
Special Light Strength 100.0%
Remaining 1/1 Light Strength 100.0%

SELF SHADOW SOFTNESS IS A GREAT WAY TO SIMULATE SOFT SHADOWS SUCH AS THOSE FROM AREA LIGHTS



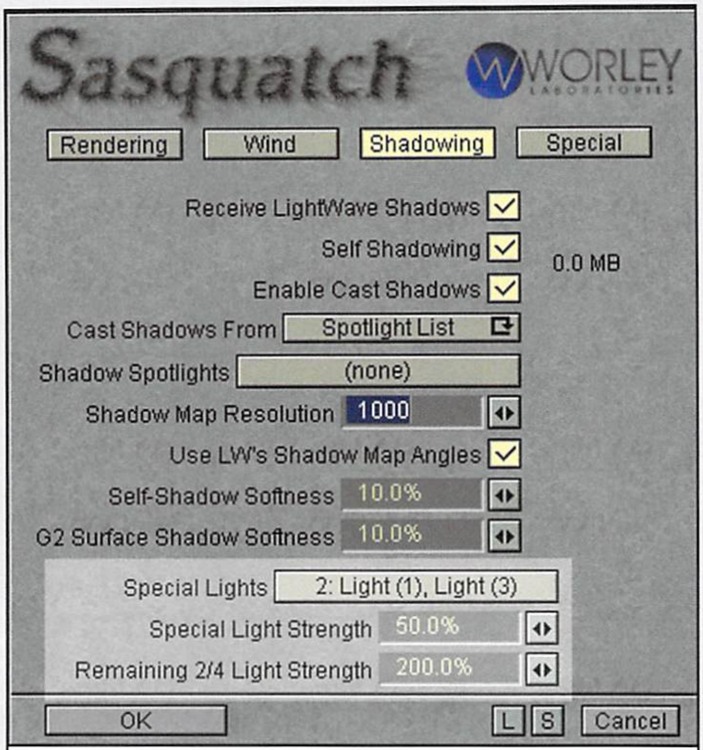
Receive LightWave Shadows ☑
Self Shadowing ☑
Enable Cast Shadows ☑
0.0 MB
Cast Shadows From Spotlight List
Shadow Spotlights (none)
Shadow Map Resolution 1000
Use LW's Shadow Map Angles ☑
Self-Shadow Softness 10.0%
G2 Surface Shadow Softness 10.0%
Special Lights (none)
Special Light Strength 100.0%
Remaining 1/1 Light Strength 100.0%

IF YOU USE G2, YOU CAN USE THIS TO SET SHADOW SOFTNESS ON G2 SURFACES

G2-SURFACE SHADOW SOFTNESS

IF you have turned on *Enable Cast Shadows* AND IF you are using G2 on your surfaces, AND IF you are casting fiber shadows onto your geometry, then this tool plugs directly into the G2 surface shader to control the softness of shadows cast from fur fibers onto geometry. That's a lot of "IF"s, but if it speeds up your process, hey, who can complain?

SPECIAL LIGHTS AND SPECIAL LIGHT STRENGTH



Receive LightWave Shadows ☑
Self Shadowing ☑
Enable Cast Shadows ☑
0.0 MB
Cast Shadows From Spotlight List
Shadow Spotlights (none)
Shadow Map Resolution 1000
Use LW's Shadow Map Angles ☑
Self-Shadow Softness 10.0%
G2 Surface Shadow Softness 10.0%
Special Lights 2: Light (1), Light (3)
Special Light Strength 50.0%
Remaining 2/4 Light Strength 200.0%

WITH THE "SPECIAL LIGHTS" LIST, YOU CAN SELECT A NUMBER OF LIGHTS TO BE "SPECIAL LIGHTS"

This is a really cool and powerful feature of Sasquatch lighting. With the "Special Lights" list, you can select a number of lights to be "special lights." You can then independently control those lights' intensity *but only on the fur!* So a light can be one intensity on the geometry and another intensity on the fur fibers. "So What?" you say. Well, if you've carefully and painstakingly textured your geometry, and then painstakingly and carefully created your fur settings, but they now don't light the same, this will help solve the problem

It will also allow you to eliminate certain lights from illuminating the fur. If you have perfect lighting in your scene, but just one light is causing you problems on the fur, you can make it a special light and set its special light strength to 0%.

REMAINING N/N LIGHT STRENGTH

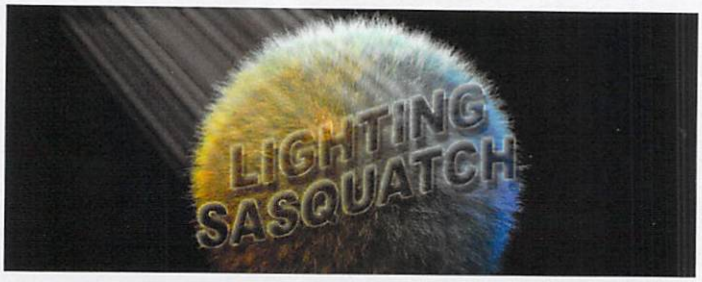
Just below the special light strength is *Remaining 2/4 Light Strength;* this means, any lights that were not included in the "Special Lights" list are considered to be "Remaining Lights." So you can also globally set the light intensity of the "Remaining" lights without affecting the "Special" lights.

I once had an array of spotlights to simulate an area light (so I could use shadow maps), and I had one set of lights for the fur and another set for the geometry. I made the geometry lights "Special" and set their intensity to 0%, and in the Geometry's Object properties panel, I excluded all the fur lights. In this way, I was able to set up two identical arrays of spotlights so that I could control the geometry lighting separately from the fur lighting.

This is just one of many ways to use these tools. As you become more familiar with them, you will certainly discover new and unique uses for them too.

Most importantly of all, remember that an experienced Sasquatch artist uses the abbreviation "Sas" instead of "Sasquatch" in casual CG conversation, ie: "Yeah, I'm gonna use Sas on that project."

Happy Sassing.



NICHOLAS BOUGHEN IS AN EMMY AND GEMINI-NOMINATED VISUAL EFFECTS ARTIST LIVING ON THE BEAUTIFUL CANADIAN WEST COAST IN VANCOUVER, BRITISH COLUMBIA. HE ENJOYS A CAREER AS SENIOR VISUAL EFFECTS ARTIST AT RAINMAKER, WORKING IN FEATURE FILMS, TELEVISION, AND COMMERCIALS. BOUGHEN HAS TRAVELED FAR AND WIDE TEACHING LIGHTING TO LIGHTWAVE ARTISTS AS FAR SOUTH AS MEXICO CITY AND AS FAR EAST AS ORLANDO, FLORIDA, WHERE HE RECENTLY PROVIDED SEMINARS TO STUDENTS AT THE D.A.V.E. SCHOOL WHO WERE WORKING ON THEIR LATEST CLASS PROJECT, *BATMAN: NEW TIMES*. BOUGHEN IS THE AUTHOR OF THE BESTSELLING LIGHTWAVE BOOK *LIGHTWAVE 3D 8 LIGHTING*, AND CONTRIBUTING AUTHOR ON *LIGHTWAVE 3D 8: 1001 TIPS & TRICKS*, BOTH PUBLISHED BY WORDWARE PUBLISHING.

Call in the
BIG GUNS
To Kill Render Times

ResPower
ResPower
Super/Farm™
Self-Service Render Farm
www.respower.com

650+ Computers, 2.5+ Terahertz
FREE TECH SUPPORT
24/7/365
3dStudio Max, Brazil R/S, finalRender Stage-1,
InSight, Lightwave, Maya, mental ray®,
V-Ray, Vue 4 Professional

Midwest NewTek Expo & Training
April 26, 2005 Milwaukee, WI

Make plans NOW to attend the Midwest's LARGEST show focusing exclusively on NewTek's VT[4] and LightWave 3D!

See the newest VT[4] and LightWave 3D features

Meet VT[4] and LightWave 3D vendors and developers who will be on hand to demonstrate their latest products, from tutorials to plug-ins and everything in between

Learn LightWave 3D with WaveGuide LIVE! From beginner to advanced, these classes taught by Pat Beck and Brad Krause will have something for you.

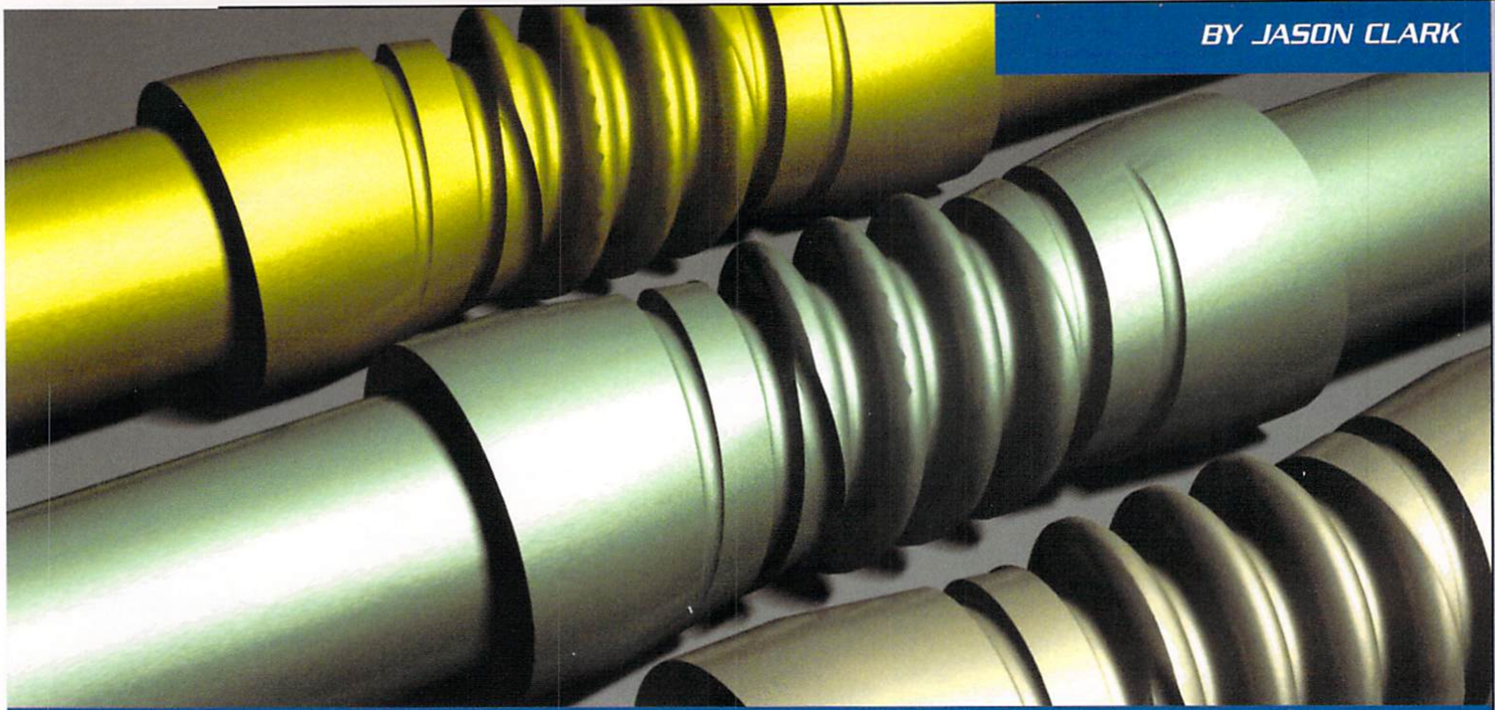
WaveGuide LIVE!

www.sharbor.com/expo

At the Clarion Hotel, Milwaukee, WI
Expo Hours: 9-5, Class Hours 8-8

Safe Harbor
800-544-6599
www.sharbor.com

BY JASON CLARK



Transferring CAD Models to DCC Applications

For many of us, the models we render don't actually come from our native application of choice; they come from our client. Many advertisements are made from digital models of actual products that exist solely in engineering software (CAD software). When dealing with CAD models rather than polygonal models derived from DCC applications, we are faced with a few issues that strongly affect the outcome of our renderings.

CAD VERSUS DCC MODELING

In the most basic of terms, there are two differences between what a CAD software user and a DCC software user will model: level of detail and geometry.

LEVEL OF DETAIL

Designers deal with details down to the tiny chamfers on an edge. When modeling in CAD software, designers are responsible for modeling any detail that may be manufactured, even more so when models are numerically control machined (CNC machining). On the other hand, a DCC artist will model what they see (typically from a camera's perspective) and what they require to make something look good or translate artistic intent. Therefore, modeling seal grooves inside an engine may not make sense for a DCC artist, but for a designer it's a required detail. When dealing with CAD models, you will find that because the models are heavily detailed, they will be extremely dense in a meshed format if not dealt with correctly. For example, it is quite typical to put a radius on sharp edges, and because this size of feature is small in comparison to the surrounding geometry, you will find that it is polygon intensive to retain such features.

GEOMETRY

The way CAD and DCC applications describe geometry is vastly different as well. CAD models are typically solids-based, utilizing feature-based modeling, B-Rep, CSG, or even a combination of these standards. In simple terms, this means that CAD models carry volume to allow designers to integrate properties such as weight, inertia, and center-of-gravity. In most cases, our rendering applications can do without this "volume" data, and as such, artists tend to model the exterior extremities of their models.

Most DCC applications are surface modelers. Typically, you will describe your surfaces with either polygons or NURBS surfaces. To mimic smoothing and NURBS surfaces, the new mainstream modeling applications utilize a method of "subdivision" and use terms like MetaNURBS, HyperNURBS, etc. "Subdee" refers to a method of smoothing a polygonal model using spline-smoothing algorithms. While DCC applications may use the term "NURBS," they are typically playing on what NURBS means to smoothing model geometry and are not utilizing the same mathematical smoothing operations.

One last issue to be aware of is that CAD models are derived by high-order B-Spline based geometry, typically NURBS. "Subdee" modeling is typically lower order spline smoothing, usually Bezier or cubic-splines. Where you find this may be a problem is in converting files that have a high order of smoothing, and as such, will translate poorly into polygons (more on this later).

THE TRANSLATION HURDLES

MODELING KERNELS AND DATA FORMATS

All applications utilize data structures and modeling algorithms differently, and extracting this information can be easy or tedious, depending on the software manufacturer. CAD software manufacturers typically utilize modeling kernels that are proprietary, like PTC's Pro/ENGINEER, Unigraphics (UGS). Until recently, neutral file formats were the only option used to translate geometry from CAD systems to DCC applications. To extract geometry from CAD systems, you had to rely on the client to create the geometry files from the CAD system, which can be a win or lose situation. Typically, CAD vendors only supported a limited volume of data exchange formats, which were typically non-polygonal.

A typical CAD system will support the following neutral formats:

- IGES (.igs, .iges): Initial Graphics Exchange Specification
- STEP (.stp, .step): STandard for the Exchange of Product
- STL: Stereolithography format (triangulated format)
- XGL: 3D information that can be rendered by SGI's OpenGL rendering library
- VRML2 (.wrl): Virutal Reality Markup Language
- ACIS (.sat)
- Parasolid (.x_t)
- OBJ: Wavefront object format (triangulated format)
- 3D DXF: Autodesk's Data eXchange Format

Since CAD vendors held the cards, you either had to interact with the client for exported geometry or you had to own a CAD system that the client used. Some clients are more giving than others; where some will just give an exported file, billion triangles and all, others will work with you in generating a clean file.

Presently, more CAD vendors are licensing their modeling kernels to allow direct access to feature geometry to assist in translations:

- *PTC* licensing Granite
- *UGS* licensing Parasolid
- *Spatial* licensing ACIS

Now, with the CAD vendors licensing their kernels, it has allowed format translation vendors like *Okino* (www.okino.com) to develop a native toolset (PolyTrans product) to use client supplied native CAD files for translation. With the native CAD file, you now have the ability to tweak the export of the files for your needs.

GEOMETRY ISSUES

We've covered a lot of ground looking at what makes CAD applications and DCC applications different and a little into what makes a CAD model unique. We'll shift our focus now to extracting the geometry from a CAD system.

Few DCC applications deal with NURB surfaces well, unless modeled natively in the application. Even SOFTIMAGE|XSI, Maya, and 3ds max have a hard time importing larger sized IGES files; it is not uncommon that a 20 or 30 MB IGES file will import very slowly, and in fact, IGES files are commonly known to reach 100MB or more, which will render a DCC application unresponsive. At some point, tessellation needs to occur to break down our smooth spline surface geometry into an approximation of triangles or polygons. For all intents and purposes, our CAD geometry is 100% accurate, and no matter what tessellation we use, we will lose that accuracy when converted to a triangle or polygon mesh.

HIERARCHY (PARENT-CHILD TREE)

When you are dealing with assemblies of parts in a CAD system, maintaining the hierarchy relationship (assembly structure) adds another complexity to our data exchange. Typically, DCC applications will not recognize a hierarchy (assembly structure) unless you use an intermediate translator, like PolyTrans, or a system of color-coded parts.

There are some formats that are able to contain hierarchy; however, a lot is dependent on the DCC application's importer. Neutral files that support hierarchy:

- IGES
- STEP
- Parasolid (.x_t)
- ACIS (.sat)

EXPORTING FILES FROM CAD SOURCE

For this section the reference software referenced:

- PTC's Pro/ENGINEER Wildfire 2.0
- Okino's PolyTrans v4.19
- NewTek's LightWave 3D v8.0.1
- SOFTIMAGE|XSI 4.2
- Cinema 4D XL 9

Typically, we can identify our translation needs as either small or large. The process that I discuss is fundamentally identical between the two needs, except in the large circumstance, we introduce the translator middleman, PolyTrans. One thing to bear in mind is that as I show the export process from Wildfire; we can substitute Wildfire (or other supported CAD) with PolyTrans in most cases, because PolyTrans supports the Granite kernel and can therefore open Pro/ENGINEER parts and assemblies natively. Obviously, if Okino does not make a native plug-in for your CAD software, then you are required to use a neutral format from your CAD software first.

EXPORTING SMALL DATASETS

I define small datasets where I am translating 1 to 50 parts where the data is quite manageable in the CAD system.

ACCEPTABLE FORMATS

Almost any format will be useful, but special considerations for maintaining hierarchy are marked with an "*":

- IGES*
- STEP*
- Parasolid*
- ACIS*
- OBJ – Wavefront
- STL – Stereolithography

If you are dealing with a heavily structured tree, I recommend using PolyTrans.

EXPORTING LARGE DATASETS

I define small datasets where I am translating 50 or more depending on the complexity of the geometry. The best formats to export are:

- CAD Native File (Pro/E, SolidWorks)
- IGES* assembly
- STEP* assembly

When dealing with larger assembly sets, I would recommend using a tool like PolyTrans. The tools within the program allow you to:

- restructure hierarchy
- tessellate NURBS surfaces
- perform polygon reduction

GENERAL CONCEPTS WHEN EXPORTING FROM A CAD SYSTEM

Using PolyTrans solves a lot of issues of translation; however, if you are faced with exporting from a CAD system, there are issues you should be aware of and understand how to handle.

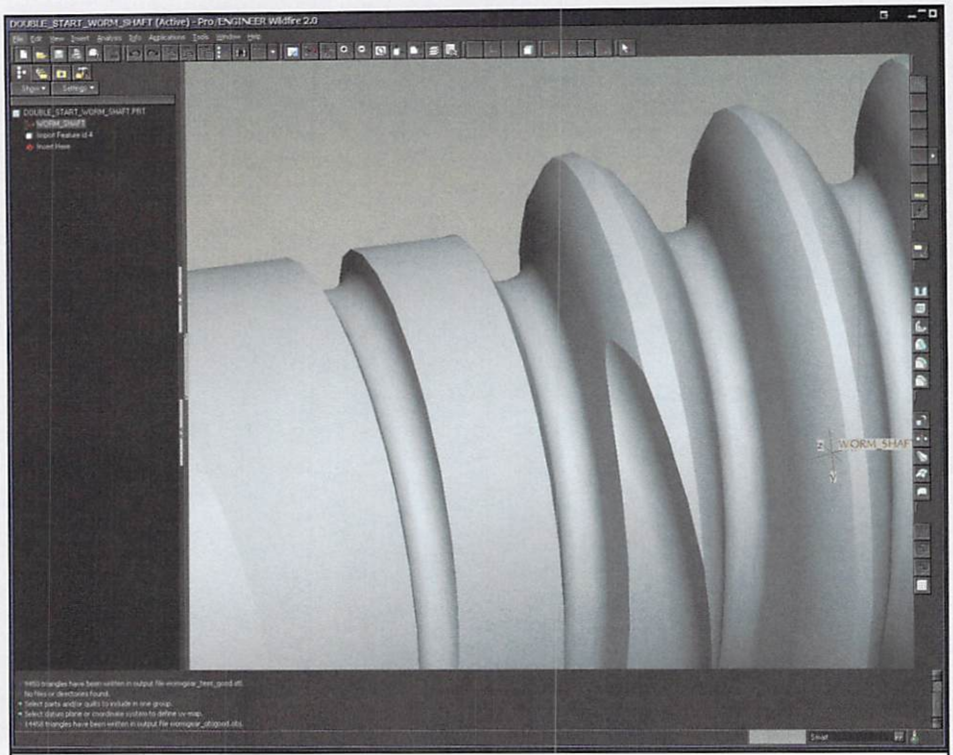
COORDINATE SYSTEMS

The equivalent of a "pivot" point or "center" in a DCC application is the coordinate system in CAD. If you want to define a center or pivot point for a model you are taking from CAD to your DCC application, you will want to define or use a consistent coordinate system. The coordinate system is especially important when you are exporting a large assembly in reduced subassemblies (whether it is for organization or size constraints). By defining a consistent coordinate system, you will find that when opening translated CAD models, they will locate themselves based on their coordinate system.

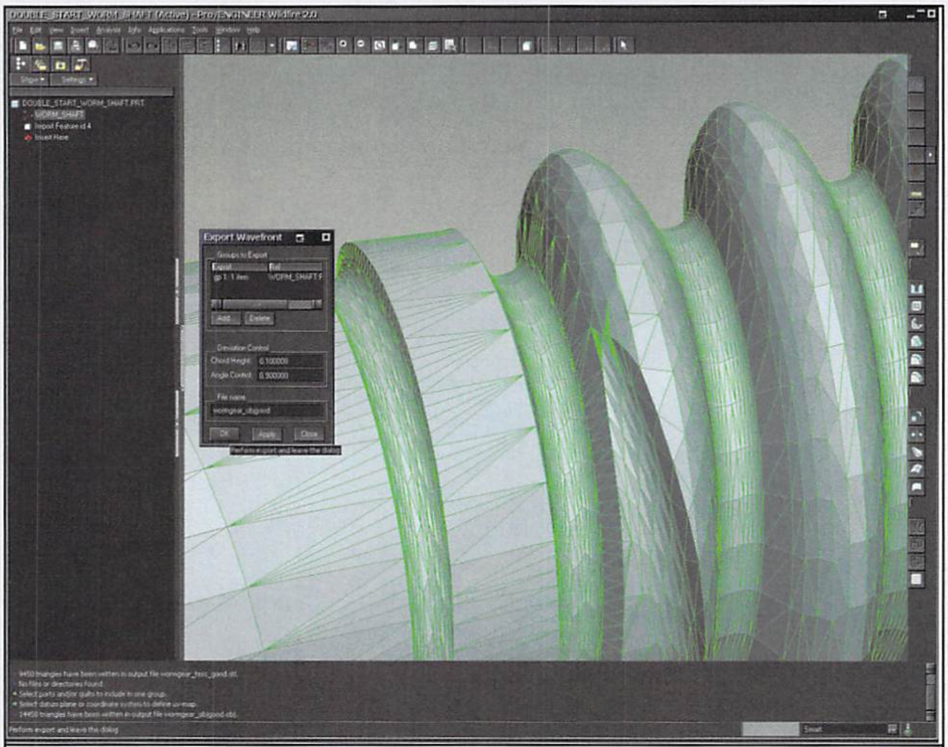
MESH DENSITY

(applicable when using STL, OBJ type export)

When setting the mesh size, you must be aware of the concept of the "bounding" box.


DOUBLE_START_WORM_SHAFT (Active) - Pro/ENGINEER Wildfire 2.0
WORM_SHAFT

EXAMPLE: WORM GEAR SHAFT MODELED IN CAD



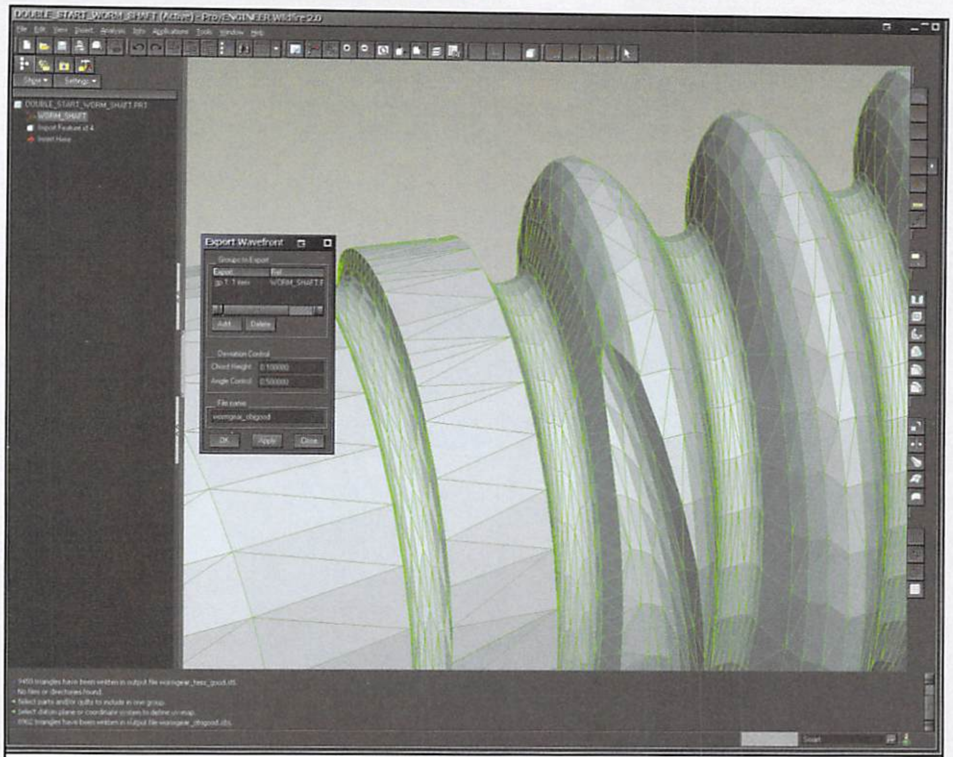
Chord Height: 0.100000

Angle Control: 0.900000

EXAMPLE: WORM GEAR SHAFT WITH A FINE ANGLE CONTROL, FINE CHORD HEIGHT

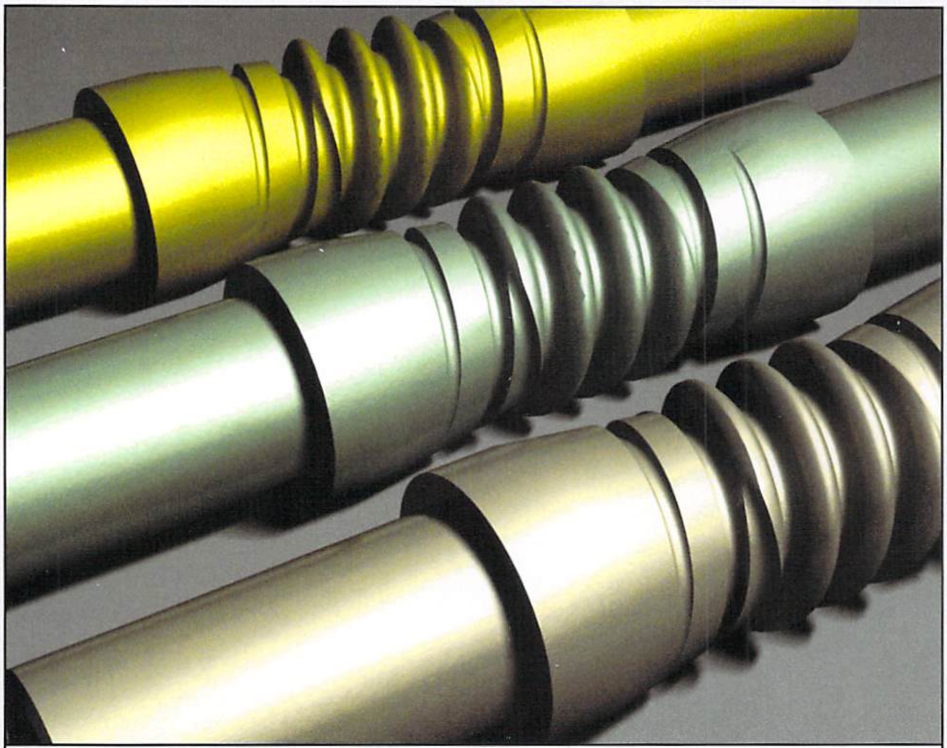
The mesh density is a formulation based on the bounding box of the model and can therefore yield extremely faceted meshes for small features (such as fillets) when the relational size of the model is very large in comparison.

Typically, a bounding box is determined by a diagonal measurement from the default coordinate system to the furthest extremity. This can pose a problem when a part or assembly is modeled "far" away from



Chord Height 0.100000
Angle Control 0.500000

EXAMPLE: WORM GEAR SHAFT WITH A MEDIUM ANGLE CONTROL, COARSE CHORD HEIGHT



EXAMPLE: RENDERED WORM GEAR SHAFT, VARIED MESH DENSITIES

the default coordinate system. Therefore, if you are dealing with a part or assembly that is rather large and not centered about the default coordinate system, you may want to center the model about the default coordinate system and take note of the offset. Apply the offset when you import the model to your DCC application.

CAD systems will generally tessellate surfaces based on two parameters (sometimes hidden behind a simplified GUI):

- **CHORD HEIGHT:** The chord height is defined as the distance between the chord and the actual surface. A smaller chord height value will result in more facets

- **ANGLE CONTROL:** Angle control regulates the amount of additional improvement provided along curves with small radii. An angle control value of 1.0 provides maximum curvature improvement.

Generally, I find that if the geometry is primarily round or cylindrical in topology, a medium coarse mesh with an angle control of 0.9 yields a good result. Setting the angle control for objects primarily square is of little use, even when dealing with fillets.

With the examples at left, you should see the difference when using a higher chord angle value with a round object. You will notice most in the fillets in the grooves.

With the render at left, you will notice the difference in mesh density where the top gear to bottom gear show a transition in modifying values. The bottom gear used twice the resolution in triangles to overcome some of the stippling you see with the two topmost gears.

IMPORTING FILES INTO YOUR DCC APPLICATION

The last step to completing our file transfer is to import the file into our DCC application. At this time, you've either received or created a neutral file from the CAD application or have used PolyTrans to open a native CAD file.

Translate the file into an acceptable format for your DCC application; again, this is where I emphasize the use of PolyTrans for the convenience of being able to export the file into a native format. If you are not using PolyTrans, convert the file into an acceptable format that your DCC application will read.

USING POLYTRANS

Utilizing PolyTrans gives me a wealth of options. Whether I'm dealing with a neutral format or a native CAD file my process is the same. While the model is open, I typically:

- Review and optimize the hierarchy
- Utilize PolyTrans' polygon reduction
- Utilize PolyTrans' polygon refinement tools

PolyTrans has tools when dealing with hierarchy files to optimize them for export, typically advantageous when dealing with IGES files.

PolyTrans' polygon reduction tool is very good in my opinion, and even utilizing an 80% reduction rate, the models render very well.

The polygon refinement tools in PolyTrans will allow me to merge triangles to leverage as many "quads" (4-sided polygons) as possible and also to create planar polygons based on some factor. Using this option will benefit you when you try to leverage the subdee options in SOFTIMAGE |XSI, Cinema 4D, or LightWave, for example. Currently in LightWave only quads are supported, so utilizing the tools in PolyTrans, I can do a lot of upfront work to prepare a model to leverage MetaNURBS, if required. Using the planar polygon processing will help alleviate the rendering errors produced by non-planar quads.

IMPORTING NEUTRAL FILES INTO A DCC APPLICATION

If you are not using PolyTrans, you will have to do a lot more manual work in cleaning up your geometry import. Some applications are better at polygon reduction and geometry optimizations than others, and unfortunately that is an article into itself. However, I will highlight:

- SOFTIMAGE |XSI has very strong poly reduction tools
- LightWave has a reasonable third-party reduction plug-in QEMLOSS 3 *(http://amber.rc.arizona.edu/lw/qemloss3.html)*
- LightWave has nifty third-party triangle to quad tool mergetrigons x *(http://www7.wisnet.ne.jp/~dprj/lw_tips/guest_p_e.htm)*

However, even with these tools, at times you may need to recreate geometry, especially if you are dealing with organic type surfaces or highly convex or concave surfaces which are hard to triangulate.

MODEL SCALE

The last area that really confuses a lot of people is the area of scale. Be aware that neutral files are unitless, but your DCC application's interface will interpret the values to their default unit system. For your reference:

- 3ds max (Discreet) – Inches
- Inventor5 (Autodesk) – Centimeters
- LightWave (NewTek) – Meters
- Maya (Alias |Wavefront) – Centimeters
- Rhino (McNeel and Associates) – Millimeters
- Pro/ENGINEER (PTC) – Inches

Therefore, if you are importing a file from a CAD system, you will want to make the accommodation for its scale. If your DCC application is metric based, you will need to apply a scale factor in your DCC application or in PolyTrans to accommodate the difference in meter and inch systems. I also find it useful to model a unit cube in the DCC application and the CAD system to serve as a double check. Unless you trust your client, you can never be sure what they are exporting, unless you are the one doing the export. Scale is most important when dealing with "real-world" kinematics and forensic type animations.

DEFINITIONS

B-Rep	Boundary Representation, a way of describing a part by its vertices, edges and faces
CAD	Computer Aided Design
CNC	Computer Numerical Control
DCC	Digital Content Creation
NURBS	Nonuniform Rational B-Splines

JASON CLARK CURRENTLY WORKS AS A SENIOR DESIGNER FOR A COMPANY SPECIALIZING IN SUBSEA INTERVENTION AND SUBMARINE RESCUE SYSTEMS, AS WELL AS AT FREELANCE DESIGN AND RENDERING. FOR THE LAST DECADE, HE HAS DEALT WITH MANY CAD SYSTEMS AND WITH INTERCHANGING DATA BETWEEN THEM. IN RECENT YEARS, JASON HAS ADOPTED RENDERING CAD MODELS FOR THE PREVISUALIZATION OF PROTOTYPE SYSTEMS HIS COMPANY DEVELOPS. HE ALSO CHAIRS THE VANCOUVER PRO/ENGINEER USERS GROUP.

BY WILLIAM "PROTON" VAUGHAN

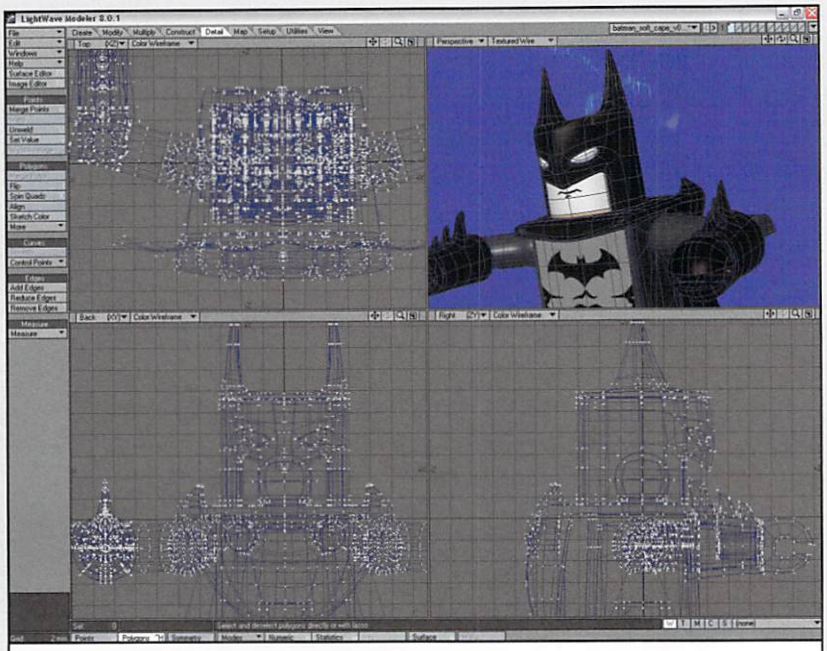
HOLY SHORTS, BATMAN!

ADAM WEST
MARK HAMILL
COURTNEY THORNE-SMITH
DICK VAN DYKE
NEW YEAR
NEW TROUBLE!
BATMAN
NEW TIMES
DAVE SCHOOL STUDENT PRODUCTIONS PRESENTS ADAM WEST IN BATMAN: NEW TIMES ALSO STARRING MARK HAMILL
COURTNEY THORNE-SMITH AND DICK VAN DYKE AS COMMISSIONER GORDON ANIMATION BY THE CLASS OF DECEMBER 2004
WRITTEN AND DIRECTED BY JEFF SCHEETZ & WILLIAM VAUGHAN DESIGNS BY ART ASYLUM STORYBOARDS MARK SIMON
SOUND BY SOUND-O-RAMA STUDIOS MUSIC BY KAYS SOUND DESIGN BY KC LADNIER VOICE DIRECTION BY MICHAEL HACK
WWW.DAVESCHOOL.COM
DAVE SCHOOL
STUDENT PRODUCTIONS
ARTASYLUM
PSYCHO TOYZ FOR CRAZY KIDZ
Sound-O-Rama
BATMAN AND ALL REALTED CHARACTERS AND ELEMENTS ARE TRADEMARKS AND © 2004 OF DC COMICS. ALL RIGHTS RESERVED.

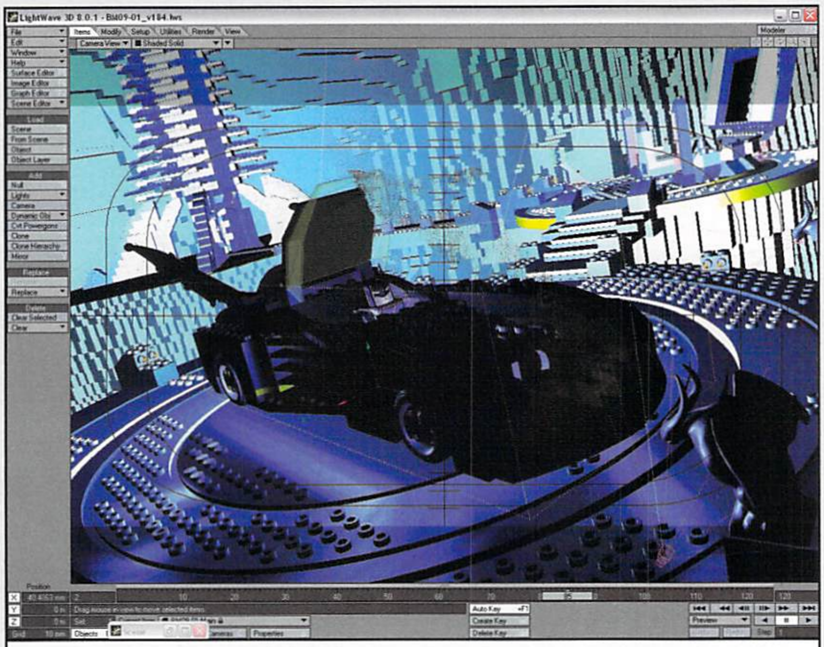
OFFICIAL *BATMAN: NEW TIMES* MOVIE POSTER

Animated short films are nothing new, but in the past couple of years the industry seems to have exploded with amazing stories told by some very talented artists. From large companies like Pixar to minnows like Out of Our Minds, animated shorts have been a proving ground for technique and development for quite some time. Recently, some of the most interesting shorts have come from schools around the world. Every three months a new short film is created by the graduating students at the DAVE school for their final project.

The December 2004 graduating class of the DAVE School recently wrapped up the production of *Batman: New Times*, a 10-minute animated short written and directed by Jeff Scheetz and yours truly. This fast-paced animated adventure follows the exploits of Bruce Wayne as his crime fighting alter ego – Batman. Bruce risks everything to deliver Gotham City from the vile Joker, while he and his henchmen do all they can to spoil Gotham's New Year's Eve party. Villainess Harley Quinn



CG BATMAN MINIMATE IN LIGHTWAVE MODELED BY ALEJANDRO PARRILLA



ALL MOVIE SETS WERE CREATED OUT OF CG LEGOS, INCLUDING THE BATCAVE WALLS

PROJECT HIGHLIGHTS

• Over 10 minutes of full CG animation •
• Over 1000 CG characters •
• Over 50 sets •

An all-star cast including: Adam West, Mark Hamill, Courtney Thorne-Smith, Dick Van Dyke, and more.

joins the fray, throwing everything she can in Batman's way.

Batman: New Times involves over one thousand characters, over fifty sets, and more than twenty vehicles. Every set, every prop, and every character was precisely modeled according to strict scale specifications. Almost every location, vehicle, and character could have been built from Lego blocks. To research this project and aid in set design, The DAVE School purchased over 100 lbs. of Lego blocks from sellers on eBay. We worked closely with Art Asylum in creating the stars of the film and based them on Art Asylum's MiniMate toy line (http://www.legionsofgotham.org/AAMiniMates.html). Art Asylum supplied us with everything we needed in the form of reference material, including the actual toys!

Working with Los Angeles-based Voice Director Michael Hack, The DAVE School assembled an all-star cast of voice talent to match the high-voltage animation of the December Graduates. TV's original Batman – Adam West – signed on early to this exciting project. Fans of the original series will recognize his signature style and over-the-top performance. Mr. West is joined by Mark Hamill, reprising his role as the Joker from *Batman: The Animated Series.* As a huge fan of the original 1960s *Batman* television series, Mark Hamill was thrilled to perform his Joker alongside his favorite Batman, Adam West! As Catwoman, we cast the talented and charming Courtney Thorne-Smith. She loved the idea of doing something her kids would enjoy. Dick Van Dyke rounds out the celebrity cast, lending his voice to Commissioner Gordon.

As with all of the DAVE School projects, we worked with well-known sound effects company Sound "O" Rama on the original score and custom sound effects. With so much talent involved on the project, a lot of pressure was on the team to pull it off. Taking on a 10-minute project with a cast of thousands in just over three months might sound crazy, but, not knowing any better, we were up for the challenge.

Now that the project has come to an end, I asked a few of the key players a couple rounds of questions in the hope of sharing an inside look at the making of *Batman: New Times.*

William: *So guys, what role did the three of you play on the project?*

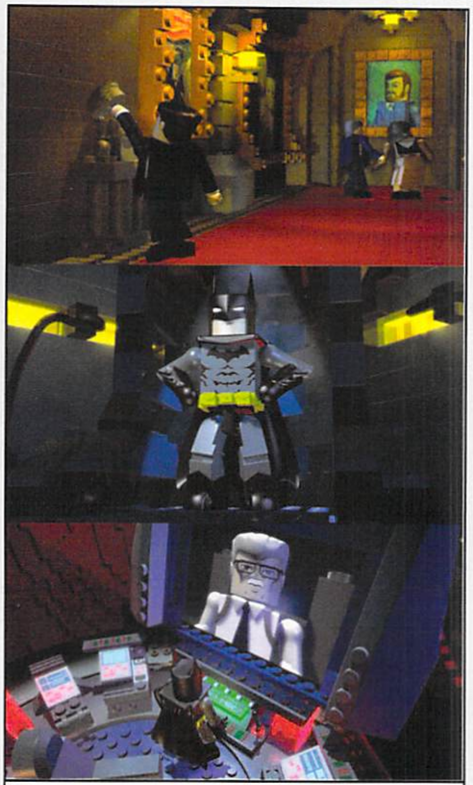
Alejandro Parrilla: *I had many roles on the production of this film. I modeled most of the main characters and some of the secondary ones. Batman/Bruce Wayne, Joker, and Catwoman are characters that actually exist in Art Asylum's MiniMates world and I had to match the action figures as they are. Then I could get creative and had more freedom to design characters like Harley Quinn and Commissioner Gordon. I also modeled, textured, rigged, and did the character animation for the entire Batcave sequence from Bruce Wayne putting on the Batsuit until the Batmobile takes off.*

Lastly, I helped to pull out some of the shots that needed some help to be finished in order to meet deadlines.

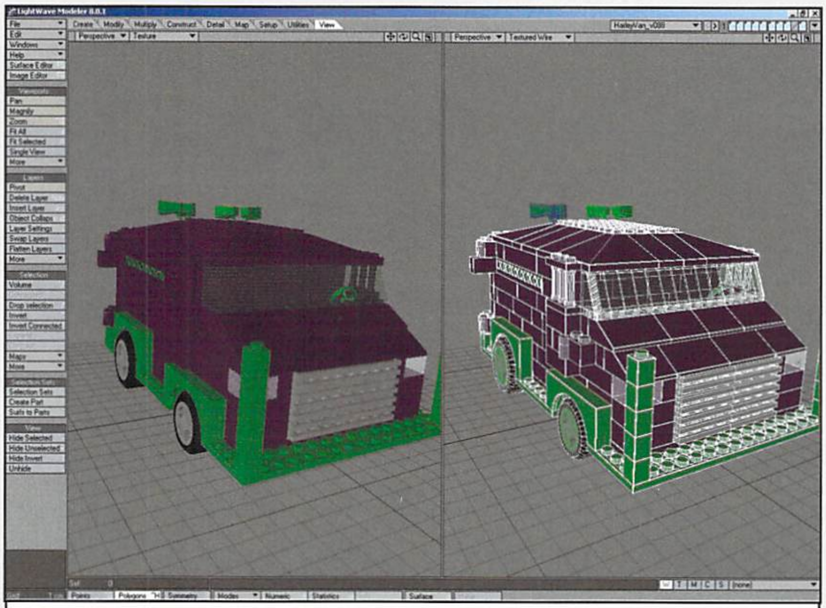
Josh Adams: *My baby in this project was the car chase through Gotham City. Since the beginning stages of the project, I was very excited about the car chase scene. It had the potential to be a beautiful-looking sequence, and I wanted to do as much of it as I could. I knew it was going to be a fun challenge that I was really hoping to take, and when I was given a major part of it by building the "Harley van," I knew that I was going to have a blast throughout the project. The coolest part about doing the van was that there was no toy or design to work off of, so I got to be creative and design the van myself. I had an "angry armored van" concept in my head, and I think I came out with a pretty good result. My next challenge is actually building the van out of real Lego.*

After the modeling stage, I had a major chunk of the car chase animation, and I ended up building the light rig that was used for the entire car chase. Working with Alejandro, we worked out a basic setup with expressions that would drive all of the

CG MINIMATES DESIGNED BY ART ASYLUM WERE USED AS THE CAST OF THE FILM



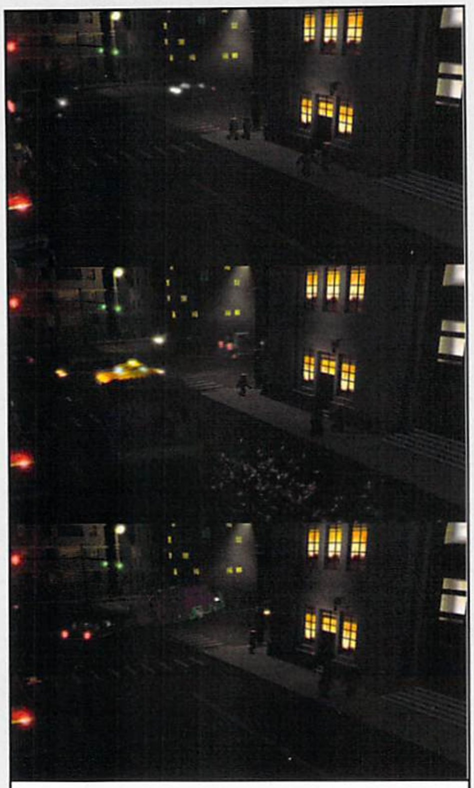
OPENING SHOTS FROM THE FILM THAT TAKE PLACE IN WAYNE MANOR AND THE BATCAVE BY ALEJANDRO PARRILLA



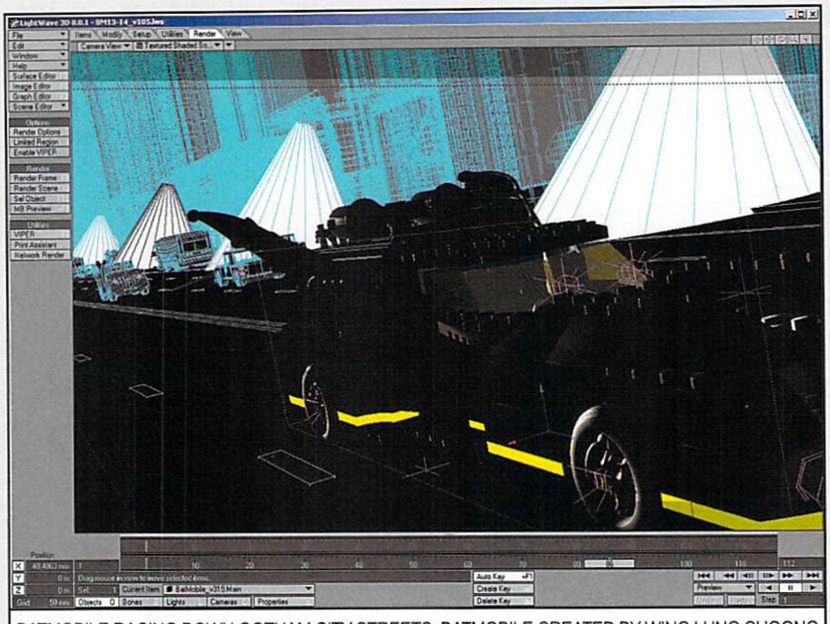
WIREFRAME VIEW OF THE HARLEY VAN USED IN THE CAR CHASE BY JOSH ADAMS

TOOLS USED

• LightWave 3D [8] • Adobe After Effects •
• Macromedia Flash • Adobe Photoshop
• Mimic for LightWave •



GOTHAM CITY STREETS CAR CHASE BY JOSH ADAMS



BATMOBILE RACING DOWN GOTHAM CITY STREETS. BATMOBILE CREATED BY WING LUNG CHOONG

cars in the city including the Batmobile and the van. But overall, the car chase was my largest contribution.

You can read how to set up the car rig here: http://www.newtek.com/products/lightwave/tutorials/animation/car_setup/index.html

Sean Fitzsimmons: *Since the project was first introduced to us, I had it set in my mind that I wanted to model the Batwing. When I was assigned to actually model it, I was extremely excited … and a little intimidated. The biggest challenge was building the model to the precise scale that we were given. Everything had to line up and match the size of the MiniMate characters that were also being built to scale.*

The next models assigned to me were the Robin and Harvey Bullock MiniMates. Once the models were finished, it was time to put our scenes together. Since I had built the Batwing and Robin, I asked to do the sequence where Robin flies the Batwing in to catch Batman after falling from the Joker-copter. Once I started to get a feel for the Batwing scenes they went pretty smoothly.

William: *Three months to work on a 10-minute animated short is quite amazing – what was the workday like?*

Sean: *The workday consisted of a few different things. Most of the time was spent focusing on whatever task was in front of me. To maintain sanity there were a few times where jokes and quotes from Adam West were thrown around with reckless abandon. And burritos … lots and lots of burritos.*

As we neared the end of the project, the nights grew longer and longer until finally the workday turned into the work night and then the work morning. Did I mention the couches (read: beds) at the DAVE School? One word. Comfortable

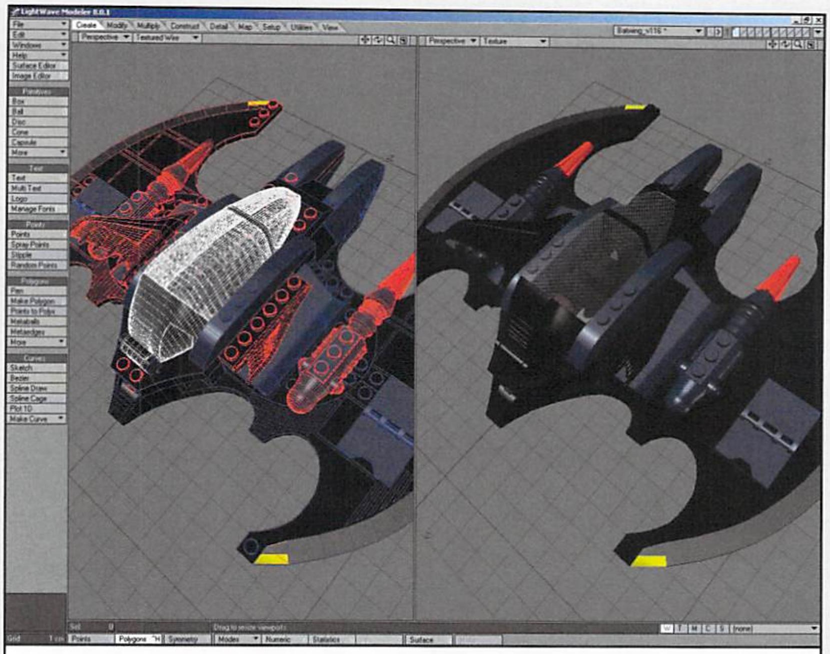
William: *How did the team handle the massive crowds in the movie?*

Josh: *The crowd shots were definitely a major thorn in the proverbial side of the Batman project. In some of the Times Square shots, there were almost 500 MiniMates in each shot, with many polygons per character, so it added up quickly. In the car chase, there were also many shots with lots of characters that got to be a burden to set up. First of all, each character had to be moving. So by making a few generic walk cycle animations and by using MD Scan and MD Plug, we were able to add movement to hundreds of MiniMates without having to animate each and every one of them, thankfully!*

Also, there were some instances where custom movements for the characters were necessary, such as in a shot where a MiniMate escapes death by jumping out of the way of the oncoming Harley van. It is little details like that that sold that shot, and sold the show entirely.

William: ***What was your favorite part of working on this project?***

Alejandro: *I actually have many favorite parts. I would probably highlight the fact that we had Nicholas Boughen as a guest speaker. He took a short break at Rainmaker and came all the way from Vancouver to share his vast knowledge with us in a two-day lighting seminar. That was unbelievable. I learned so much in that short period of time.*



SEAN FITZSIMMONS RECREATED THE ART ASYLUM BATWING TOY PIECE BY PIECE

Sean: *Modeling the Batwing was absolutely my favorite part of the project and I learned so much doing it.*

William: *Josh, you got to "co-star" with Adam West playing the role of Robin. What was that like?*

Josh: *Oh, you mean A.W.? He and I are best friends now. He flies me out to his house in L.A. all the time, and we play X-Box and sip margaritas while floating in his pool…Well, actually, I never got to meet him, unfortunately, but the whole experience was cool. Listening to a seasoned veteran of the acting world do his craft, although over the phone, was an experience I won't soon forget. I mean, come on, he IS Batman. The recording process for me was also lots of fun. I had been to recording studios in the past as a musician, but never to as nice place as Sound "O" Rama. They have a great set-up, and a very professional atmosphere. I also got a laugh out of being referred to as "talent."*

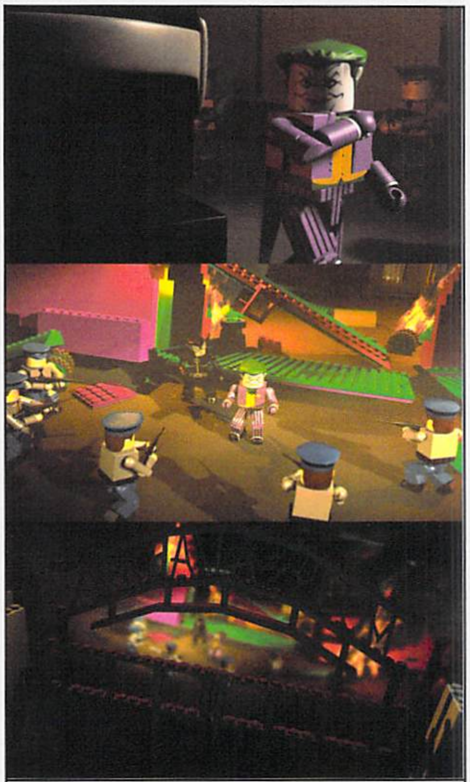
William: *Alejandro, You spearheaded bringing Mimic into the pipeline for lip sync – can you tell us a little about using it in production?*

Alejandro: *Well, at the beginning it sounded like it was going to be a hassle to add a brand new software to the toolkit, as we didn't have much time for research and development, but once you open Mimic in the LightWave interface, you find out it is rather intuitive and the steps for the whole lip sync process are pretty straight forward. It took me just a few hours to read the documentation and I was ready to go. I look forward to using it on future projects, as I know it will make quick work of any lip sync work thrown my way.*

William: ***Were there any challenges during the making of the film?***

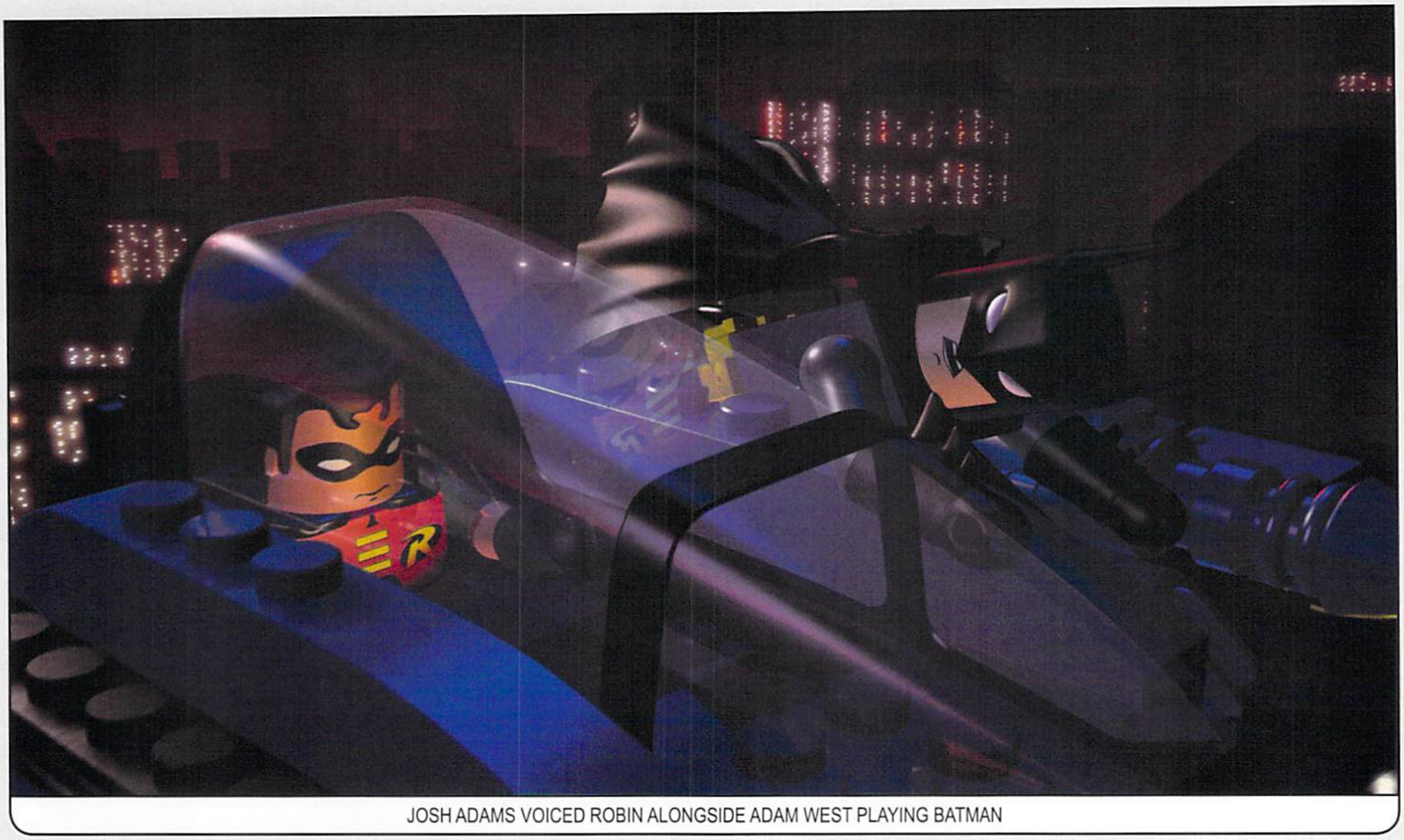
Josh: *I guess the biggest challenge of the project was working with very specific Lego designs. Every single model of the project had to be re-creatable in the Lego universe using standard bricks, so that threw us a lot of curves when trying to decide how something would look. Something that would be relatively easy to model was extremely difficult to recreate using Legos.*

Another challenge was the massive size of some of the scenes. In one of the car chase scenes we had 30 Lego buildings, 25 cars, 150 MiniMates walking on the sidewalk, the Batmobile, Batman, the Harley van, Harley Quinn, her driver, particles, street lights, and a fire hydrant – a pretty massive scene that had to be broken up, but we pulled it off and it looks great. Lighting was also a major concern in this project. We were having difficulty with how to pull off the nighttime look while being able to see all of the action, so I emailed Nick Boughen and asked for his input on some renders we had done, and with his help I feel we pulled off the nighttime street look beautifully. I just hope he feels the same way …



SEAN FITZSIMMONS AND LARRY MOORE ANIMATED THE FINAL SEQUENCE IN JUST UNDER THREE DAYS

Sean: *A big challenge for me in this project was working with particles. This was all experimentation for me and at one point I was very close to just giving up. I packed up for the night and spent the night going over the experimentation of the previous six hours in my head. Somewhere in the middle of the night the solution hit me. Trumpets blared and I was engulfed in*



JOSH ADAMS VOICED ROBIN ALONGSIDE ADAM WEST PLAYING BATMAN

light. I came to school the next day and within ten minutes had the problem (a missile trail) resolved.

William: *Any tips you learned during the production of the film that you would like to share?*

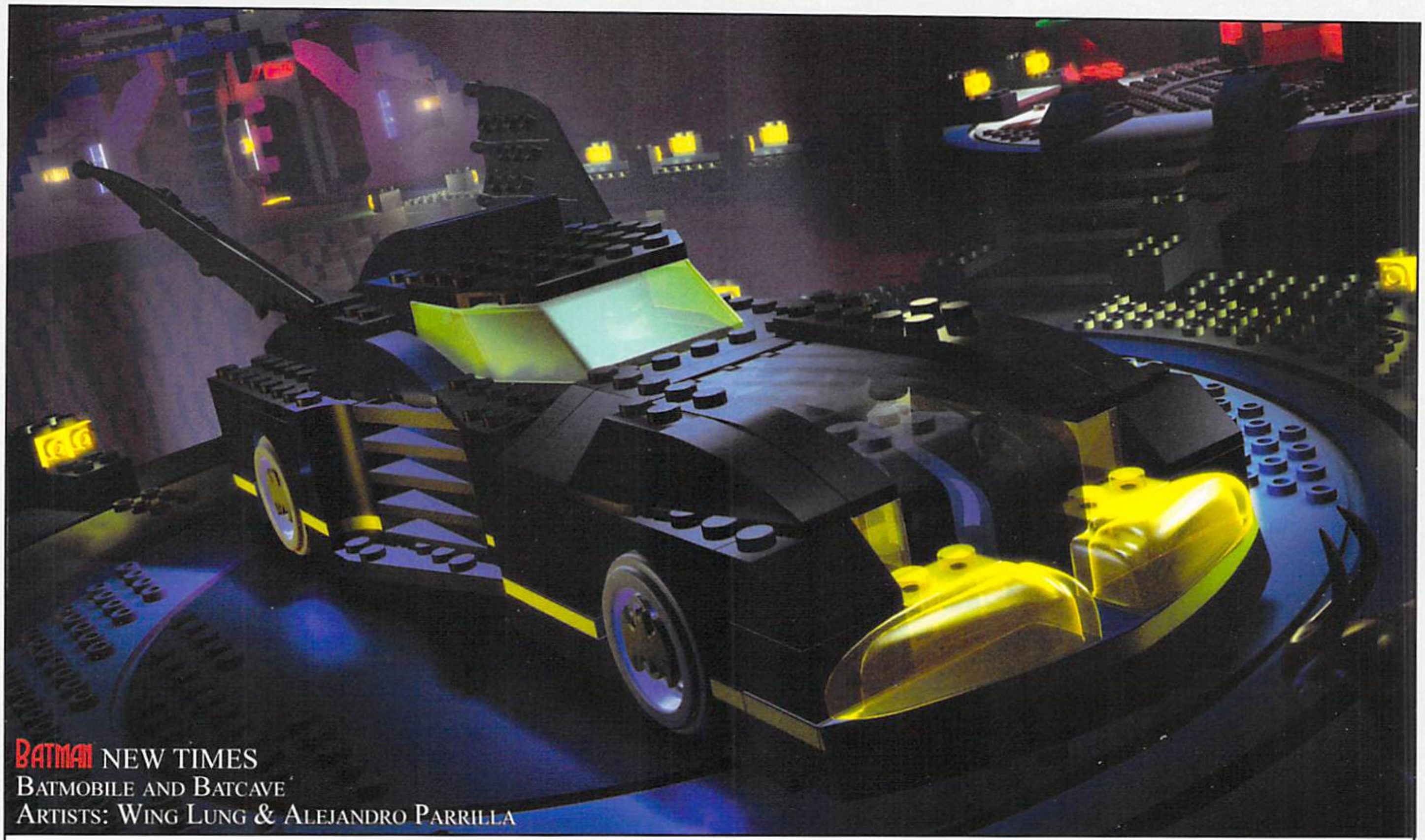
Alejandro: *One day I was trying to figure out the best way to make the model of the Batcave – try to imagine modeling the bumpy walls of the Batcave out of Lego pieces. It would take forever and the polygon count would make the scene impossible to render in short time. I got a nice tip to model using dynamics in Layout from Jonny Gorden. I wanted to recreate a sub-patch model I had designed, translated into the Lego world, so all I needed was to use the sub-patch version of the Batcave as a collision object, and a grid of unwelded polygons to match the size of the Lego blocks. By applying hard body dynamics to this object and making it fall vertically on top of the collision object with the proper settings, the grid would get the shape of the Batcave as it falls and collides, letting you save the end result as a transformed object. Then all you have to do is clean up the geometry back in Modeler, extrude the grid, and voilà! You got a Lego Batcave wall. Thanks for the tip, Jonny!*

Sean: *The biggest tip I can think of came while I was modeling the Batwing. Once the model was done and I went to do some renders of it, I found that the curves in the wings didn't have enough segments and looked too faceted. I was getting nervous thinking I was going to have to redo the wings. That's when I learned about a new tool in LightWave [8] called Edge Bevel. I went around the wings and at each of the segments, I selected the edge that was too sharp and applied the Edge Bevel. This split the edge into two, rounding out the edges of the curves. Very handy tool, and it saved me from remodeling the wings.*

William: *Now that you have experience producing an animated short, what advice would you give to someone wanting to work on one?*

Alejandro: *I would encourage people to go through the process of making an animated short. My best advice is to establish a structure so you can sail smoothly through the whole process. Establish deadlines on a daily basis. Be organized. The more disciplined you are the better the result will be. Use references. Always try to keep getting feedback from other artists; otherwise, you could be at the risk of falling into a labyrinth, which could take a while to get out from, and you may miss your deadline.*

Sean: *Organize. One thing I started doing toward the end of the project was keeping a "diary" of shots for the day. I used that to keep track of what I had done the day before and was ready the next day to check the renders. The shotlists we were given also added to the organization. We could mark off what was animated, lit, approved, and finally rendered. Another important piece of organization was our directory structure and naming convention. Working with 20 people on a project really drives home the*



FINAL BATMOBILE AND BATCAVE CREATED BY WING LUNG AND ALEJANDRO PARRILLA

need to name your scenes in the same way and keep things organized so that anyone in the group can pick up your scene and continue working.

Josh: *Well, I guess the best advice I can give could be applied to any project, whether short term or long, is to set short term goals so that you get the project done in a timely manner. The hardest part of doing this project was finishing it. So I guess I could say to see the project through to the end. As William would always say, "The last 10% of the project is 90% of the work," and I am now an advocate of his wisdom.*

Working on a Batman animated short and working with an all-star cast was amazing. But nothing can ever really compare to sitting back and watching a movie that you had a hand in creating. I would like to encourage everyone to work on an animated short at some point in their career. You will take away far more experience then you could ever imagine. I hope you enjoy our film!

FROM LEFT TO RIGHT, ALEJANDRO PARRILLA, JOSH ADAMS, AND SEAN FITZSIMMONS

To find out more about *Batman: New Times,* visit: http://www.daveschool.com

To see a behind the scenes photo montage created by the class, download the QuickTime movie from here (20MB): ftp://ftp.newtek.com/multimedia/movies/Slideshow_2004.mov

WILLIAM "PROTON" VAUGHAN
A RECIPIENT OF SEVERAL NEW MEDIA ADDY AWARDS, WILLIAM HAS AN EXTENSIVE BACKGROUND IN CREATIVE 3D FOR PRINT, WEB, MULTIMEDIA, GAMES AND BROADCAST. DURING THE LAST 10 YEARS, HE HAS DELIVERED AWARD-WINNING WORK FOR CLIENTS SUCH AS COMPAQ, NEW LINE CINEMA AND HALLIBURTON. WILLIAM HAS ALSO TRAINED ARTISTS AT SEVERAL STUDIOS AND SCHOOLS AROUND THE WORLD AND CONTRIBUTED TO SIX LIGHTWAVE 3D BOOKS THROUGHOUT 2003 AND 2004.

IN 2002, VAUGHAN JOINED NEWTEK'S MARKETING TEAM AS THE LIGHTWAVE 3D EVANGELIST, WORKING CLOSELY WITH THE LIGHTWAVE DEVELOPMENT TEAM, KEY ACCOUNTS, AND THE GROWING NUMBER OF END USERS TO ENHANCE LIGHTWAVE'S FEATURES SET.

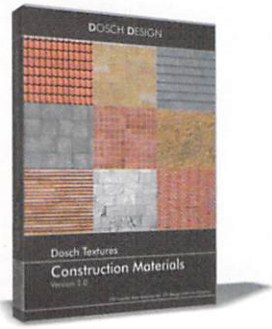
WILLIAM IS DIRECTOR OF INDUSTRY RELATIONS AND INSTRUCTOR OF THE DAVE SCHOOL 'S UPCOMING FINAL PROJECT. WILLIAM'S FOCUS IS ON CONTINUOUSLY IMPROVING THE QUALITY OF EDUCATION AT THE DAVE SCHOOL, WHILE FURTHER ESTABLISHING THE SCHOOL'S PRESENCE IN THE INDUSTRY.

DOSCH DESIGN

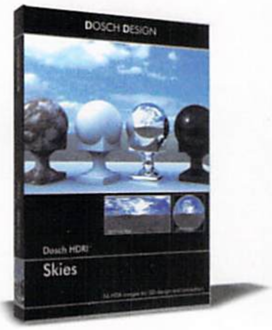
Dosch 3D: **Animated Humans for Cinema4D**

Dosch 3D: **Interior Scenes**

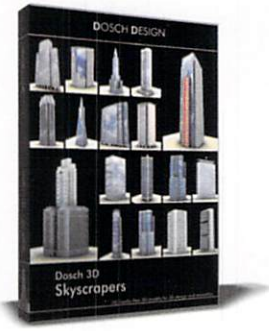
Dosch LayerFX: **Architecture**


Dosch Textures: **Construction Materials V2**

Dosch HDRI: **Extreme Hires**


Dosch HDRI: **Skies**

Dosch 3D: **Logo Animations for 3ds max**

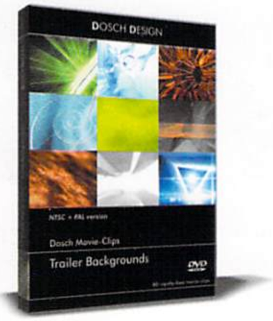

Dosch 3D: **Skyscrapers**

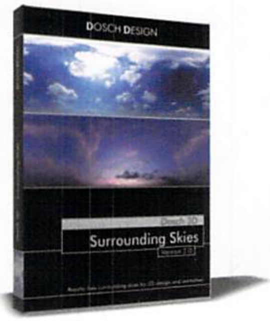
Dosch 3D: **Packaging Design V2**

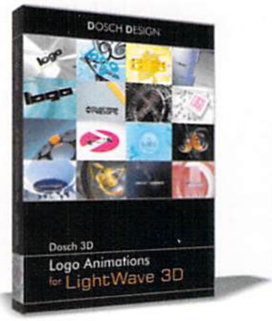
Dosch 3D: **Virtual Sets**

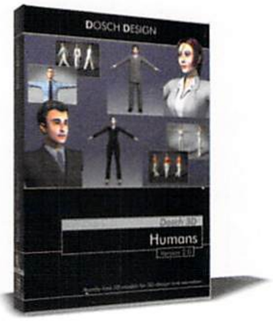
Innovative and *easy to use* computer graphics products:

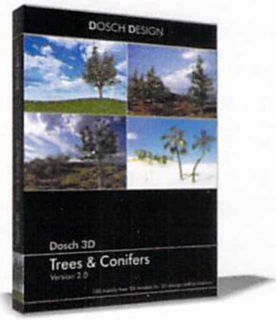
Dosch 3D: **Electronics**


Dosch Movie-Clips: **Trailer Backgrounds**


Dosch 3D: **Surrounding Skies V2**


Dosch 3D: **Logo Animations for LightWave3D**


Dosch 3D: **Humans V2**


Dosch 3D: **Trees & Conifers V2**

Dosch 3D: **Shop Design**

3D-Design, Animation, Visualization

Besides complete 3D-models and scenes **➤Dosch 3D**, Dosch Design products contain surface materials **➤Dosch Textures**, High Dynamic Range Images **➤Dosch HDRI**, as well as 2-dimensional objects for architectural visualizations **➤Dosch VizImages**.

Animated movie sequences **➤Dosch Movie-Clips**, plus quality music and sound effects **➤Dosch Audio** complement this product segment.

Graphic Design, Desktop-Publishing, Webdesign

➤Dosch LayerFX products unlock a new realm of creative freedom - and therefore substantial time and cost savings.

Order your free Info-Pack!

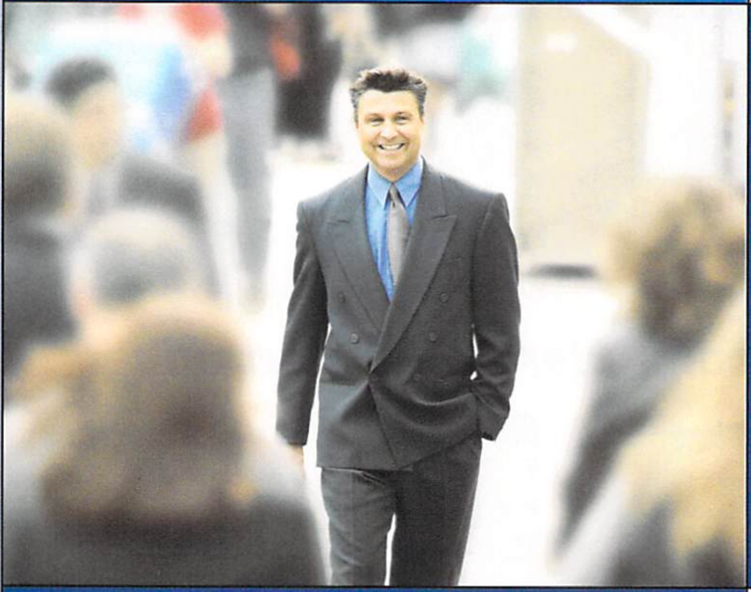
www.doschdesign.com

DOSCH DESIGN US-Shipping Center
603 Ruskin Drive - Altoona, PA 16602 - USA
Phone/Fax: ++1 (814) 943-2807
1 (866) 3D-DOSCH (US/Canada Toll-free)

DOSCH DESIGN Europe
Gotthard-Schuell-Str. 14 - 97828 Marktheidenfeld - Germany
Phone: +49(0)9391-915853 - Fax +49(0)9391-915854
info@doschdesign.com

BY ED HARRISS

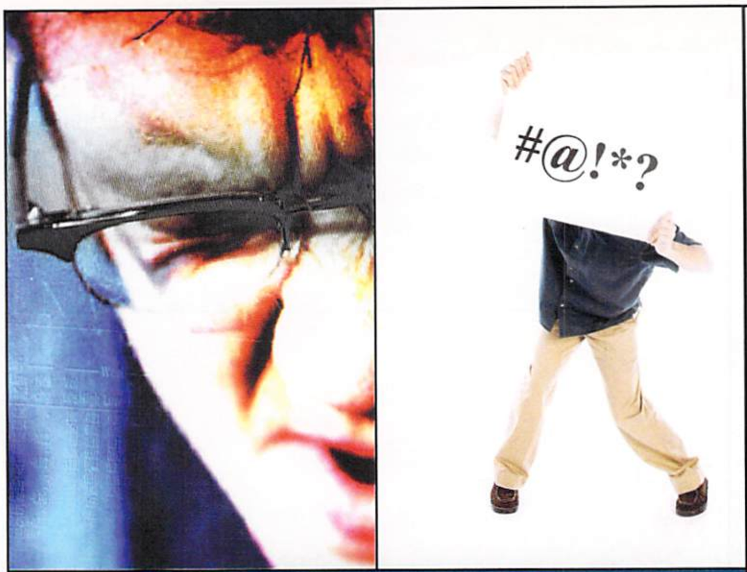
Computer Animation Job Hunting Tips —



Gone are the days when it was easy to get a 3D job. It's an employers market now. Applicants really have to stand out in order to get hired. In this article, you'll find dozens of tips on how to get noticed in today's crowded computer animation industry.

IN ORDER TO OBTAIN A JOB IN THIS INDUSTRY YOU HAVE TO FIND A WAY TO STAND OUT FROM THE CROWD.

Things not to do: Often it seems that many job hunting CG artists need to know better what *not* to do rather than what *to* do. For example, don't bad mouth your previous employer, no matter how much you might hate them. Leave satanic, erotic, and violent material off of your demo reel. Do not put color bars and sample tone at the beginning of your reel. Do not ask for your reel to be returned after it is viewed. Never send out the original copy of your reel. Don't send the same reel in over and over. (Animators have a very good visual memory and will notice.) No "works in progress." If it is not finished, don't put it on your reel. Don't put tutorials or tests on your reel unless you have drastically altered or improved them. Don't show any "plug-in animations" either. That is, don't put anything on your reel that only looks interesting because you used some "cool new plug-in" to make the effect. This should almost go without saying, but never put other people's work on your reel unless you worked on the project with them, and NEVER put work on your reel that you had nothing to do with. If you are sending out VHS tapes, always rewind the tape before you send it out. Be kind, please rewind.



IN ORDER TO OBTAIN A JOB IN THIS INDUSTRY YOU HAVE FIND A WAY TO STAND OUT FROM THE CROWD.

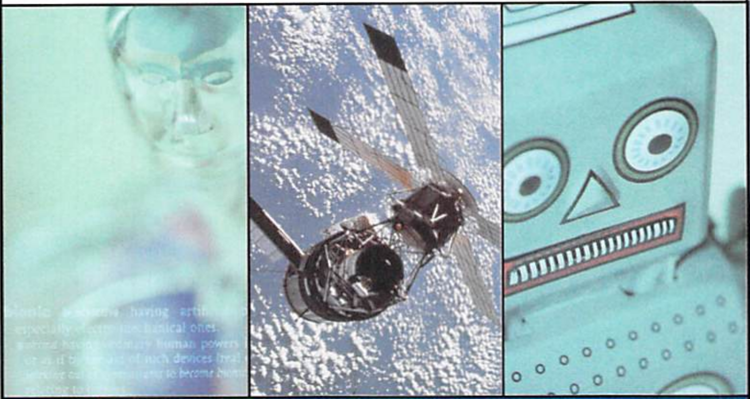
Reels: Your reel is one of the most important tools you'll use to get work. Consider it your calling card. Most demo reels must grab the interviewer's attention in the first 30 seconds or they're not going to look at the rest. (They are probably watching it during lunch and don't have time.) If they don't like what they see, they'll fast-forward the reel, running the risk of missing something good. Keep the length under 3 minutes, even if you've got 15 minutes of great work.

WITHOUT AN EYE-CATCHING DEMO REEL YOUR FUTURE EMPLOYER WILL SPEND MORE TIME WITH THEIR HAND ON THE FAST FORWARD BUTTON THAN THEY WILL WATCHING YOUR MATERIAL

FF
PLAY
VCR/TV DISPLAY LIGHT
INDEX
ADJUST
Q.V.
TV
VCR
MUTE
REC
STOP
PAUSE
REW
PLAY
FF
INDEX
VCR/TV DISPLAY LIGHT
MUTE
STOP
PLAY
REC
REW
INDEX
ADJUST

It's very important that you put your best work at the beginning of the reel. Never put anything on your reel that you have to think twice about. If you are going to have to make excuses for the poor quality of a certain segment, leave it off. Bad work often stands out more than good work. Make sure that your work demonstrates your abilities and creative talents. It needs to show that you have the ability and skills to do the work. It's not a good idea to repeat any segments on your reel. It will only look like you are trying to meet some sort of artificial "time requirement."

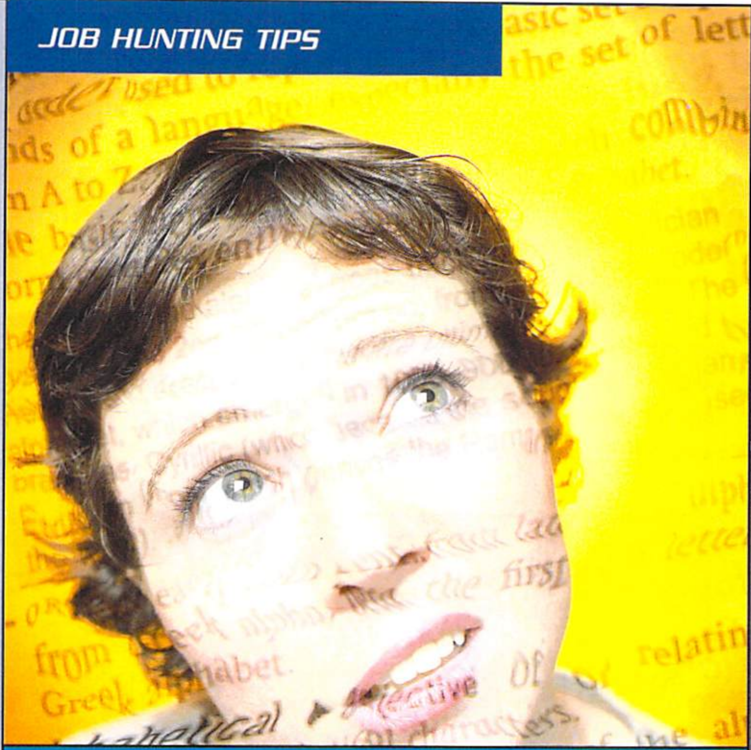
Creativity: It's a good idea to show creativity in your work. Your reel will get a lot more respect if it contains new and original work, rather than another *Matrix* rip off. You are not the first one to animate a space battle or dinosaurs or robots or tanks. If you are a student, or you are making your demo reel at home, then chances are you can do whatever you want. Pick something that you enjoy, something that other people have *not* done over and over. This expression of creativity has the potential to impress the viewer and help your reel stand out from the crowd. The easier it is for them to remember your work, the better your chances are of getting hired.



TRY TO BE ORIGINAL. NO MORE SPACE BATTLES, DINOSAURS, ROBOTS OR TANKS. DEMO REEL REVIEWERS ARE GETTING VERY TIRED OF SEEING THEM

Creativity is good not only in the animation, but in the way it is put together. Make your reel unique with some interesting editing, packaging, and music. It will really help you stand out. Don't spend an obscene amount of time on this area, but don't dismiss it entirely either.

Portfolio: If you are a good 2D artist (drawings, paintings, photography, etc.), include prints of your artwork with your reel. If that is not possible, then scan in your work and include it on your reel. Same goes for 3D work, such as sculptures and installations. Companies are more likely to hire a good modeler that can also sculpt, as opposed to someone who can only model in 3D.



WHICH TYPE OF SCHOOL IS GOOD FOR YOU?

Education: Education is a very important step in getting a computer graphics job. There are basically three educational paths available to you: attending a university, attending a specialized school, or teaching yourself. Everyone has different needs when it comes to education. Which one is right for you?

A university education will typically consist of a broad education where you will learn many things that have nothing to do with your future job. Believe it or not, that is one of the main benefits of attending a university. When you graduate, you will be a well-rounded individual that will be able to tackle a wide variety of problems relatively easily. Another benefit is that you'll have a long time to develop your skills, as typical university degrees take at least four years to earn.

UNIVERSITY EDUCATION IS SO BROAD THAT IT OFTEN PRODUCES ANIMATORS WITH WELL ROUNDED SKILLS

A specialized school will consist of a more focused education where you will learn exactly what you need to hit the ground running the instant you graduate. These schools also tend to react faster to hardware, software, and industry changes. As a result, your education will probably be a little more "up to date" than a traditional university education. Schools like this will also offer you more "hands on" instruction. Since the classes tend to be more focused, there will probably not be any history, language, or philosophy classes. These schools are also an easy way for people who already have a job to change careers quickly or refresh their skills.

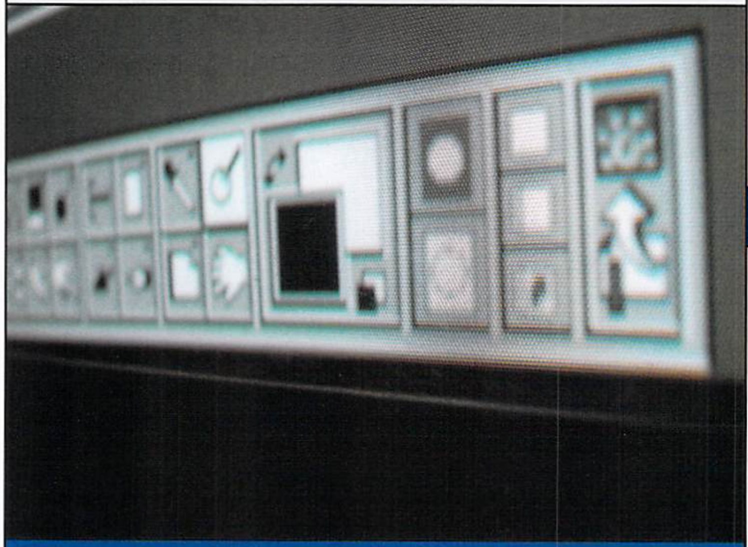
SPECIALIZED SCHOOLS OFFER A MORE FOCUSED EDUCATION

What about teaching yourself? It can be done, but you have to be very dedicated to do it well. If you are the type of person that needs someone standing over your shoulder or someone giving you deadlines in order to get things done, then the self-taught route is not for you. However, if you are good at scheduling your time, able to learn from reading manuals (rather than having some one show you), and you have the dedication and the equipment, then the self-taught route might just be for you. Many people go the self-taught route and flake out after a few months. Don't let this happen to you. If you even *think* that it might, then go to school instead. There is something to be said for having others around you when you are trying to learn CG. If you go the self-taught route, try applying at smaller companies first. They may have fewer formal educational requirements and may be more willing to zero in on your knowledge and skills rather than where you acquired them.

TEACHING YOURSELF COMPUTER ANIMATION CAN BE A DAUNTING TASK

None of these three routes are "better" than the other. In fact, some of the best CG artists out there are often a result of a combination of these three educational paths. Different people have different needs. There are pros and cons to each of the three types. The university route will take you longer and might not give you the specific knowledge that you need to know, but you'll have more time to develop your demo reel and you'll learn things that you'd never learn in a specialized school due to time constraints. Going to a specialized school might increase the chances that you'll get a job immediately after you graduate, but the broad range of knowledge that you'd gain at a university might be more important to you down the road. If you go the self-taught route, you'll save a lot of money and you'll be able to learn exactly what you want, but without structure and interaction with other CG artists, you might not be able to learn what you need to become good enough to get a job.

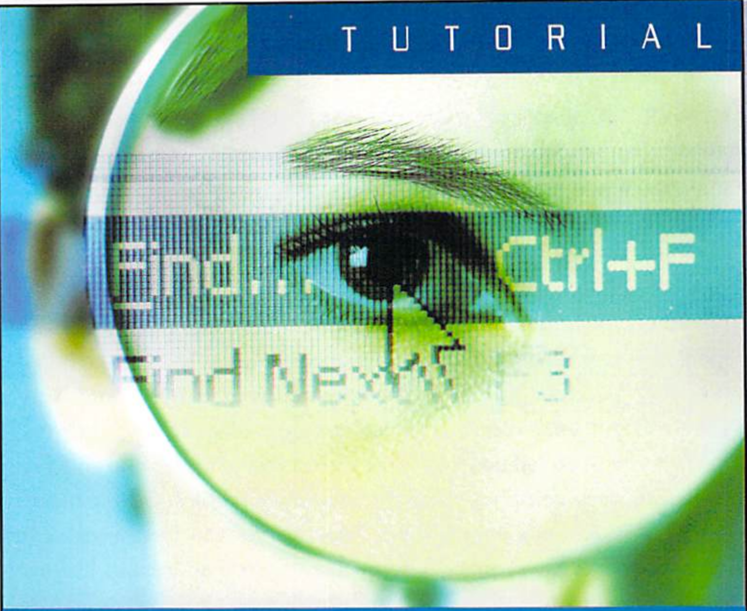
Tools: Some students get a little too obsessed with what they might think is the best animation tool. There are hundreds of companies that are using their own proprietary software. Students will probably not learn the exact tools they'll need when they enter the industry. It is best to learn the software that you can get your hands on. A student might learn SOFTIMAGE|XSI, 3ds max, LightWave, or Maya as their main software. While this is a great idea, most companies are looking for people who are good at solving problems, not just pushing buttons.



DON'T BECOME OBSESSED WITH A PARTICULAR SOFTWARE TOOL. LEARN THE BASICS FIRST

RESUMES AND COVER LETTERS:

Unless you've had dozens of jobs in a very short time, don't use your resume to explain why you left a job; leave the explanations for the interviews. Employers hate it when they have to guess what you are trying to say in order to decipher your intent. Avoid rarely used or very large words. Don't write complicated and confusing sentences. Cover letters and resumes are not excuses for you to show off your storytelling ability or vocabulary knowledge. Keep the information concise and straight to the point.



SPEND A LITTLE EXTRA TIME ON YOUR RESUME. IT MIGHT BE MORE IMPORTANT THAN YOU THINK

Resumes take longer to write than you might think. It is also difficult to come up with all the relevant information instantly. Spending a small amount of time on a resume every day for a week is much better than trying to write it in one sitting. This way, it will be much easier to add in things as you remember them.

Be a Team Player: No matter how good you are, you won't last in this industry unless you can work well with others. The ability to share information with your co-workers and get along with them at the same time is very important.

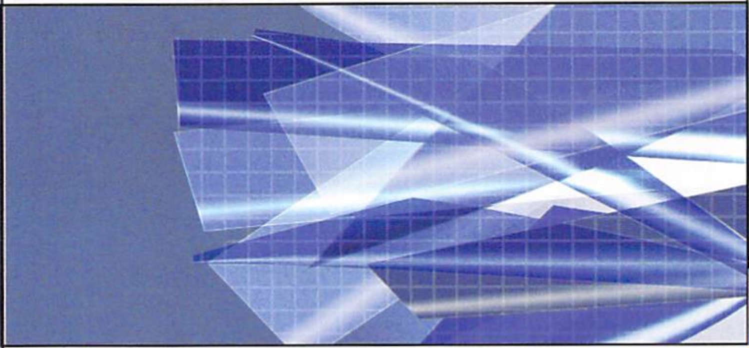
BEING A TEAM PLAYER IS VERY IMPORTANT IN THIS INDUSTRY

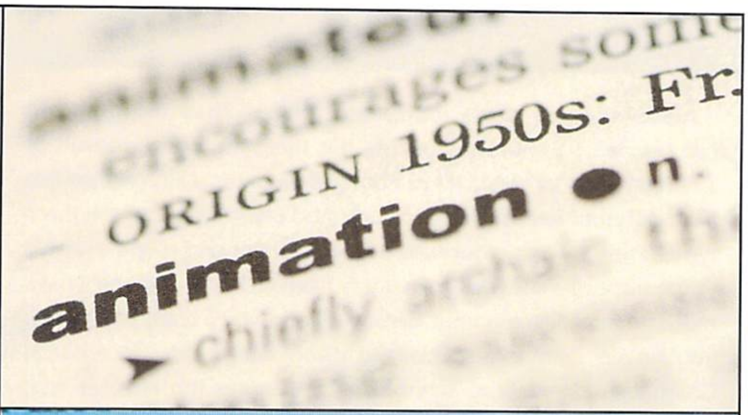
Getting your reel and resume into the right hands: All of this hard work will be wasted if no one sees it. There is a better chance that a potential future co-worker will see your work at a smaller company, as larger companies have a human resources (HR) department that you need to get through first.

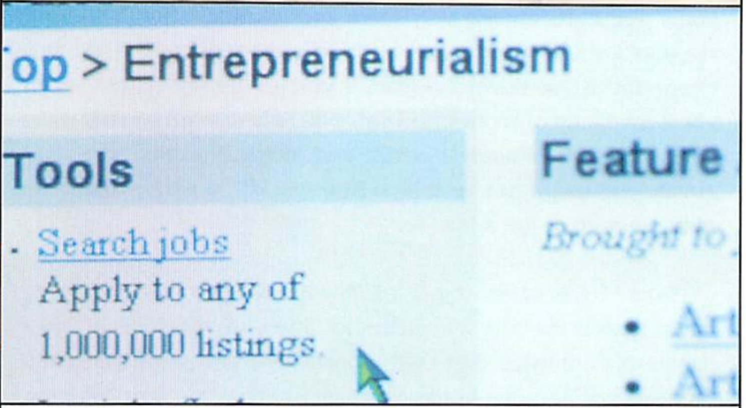
When you send in your reel to a company who is advertising for jobs, HR is where it typically ends up. They usually have so many reels to get through that they might not even get to yours. You need to get your reel to a hiring manager or a "real" employee. When you see a job posting, go to the company's Web site and get a name. Many company sites have employee profiles. Get the name of whomever the open position is likely to report to. Once you have their name, it will probably come with an email address. If it does not, a little creative Web searching usually turns it up.

YOU'LL HAVE A BETTER CHANCE OF GETTING HIRED FOR A JOB IF YOU CAN FIND CONTACT INFORMATION FOR WHOMEVER THE OPEN POSITION IS LIKELY TO REPORT TO

Once you've got their email address, ask them if it is OK if you send them your reel. When they get your reel, they will either watch it themselves or give it to HR. Why go through all this trouble when HR might get the reel anyway? Because the HR department is looking at a reel that came to them from an internal source, not just another outside submission. This makes a huge difference in their eyes.







op > Entrepreneurialism

Tools

- Search jobs
Apply to any of 1,000,000 listings.

Feature

Brought to

- Art
- Art

The bottom line: No one cares what school you went to or how cool your computer is. All that matters is the quality of your work. If you are a good CG artist, then you'll get a job. Don't worry if it is taking longer than you expected to find a job. It only takes one, and there are thousands of them out there.

ED HARRISS HAS BEEN WORKING PROFESSIONALLY IN THE COMPUTER ANIMATION INDUSTRY FOR OVER 10 YEARS. IN THAT TIME HE HAS WORKED IN MANY OF ITS DIFFERENT SECTORS: VIDEO GAMES, SPECIAL EFFECTS, COMMERCIAL PRODUCTION, CORPORATE ANIMATION, TECHNICAL VISUALIZATION, MEDICAL ANIMATION, WEB GRAPHICS, ANIMATION INSTRUCTION, TRADE SHOW GRAPHICS, MAGAZINE ADVERTISING, AND MUCH MORE. HE HAS GIVEN LECTURES AND TAUGHT CLASSES AT MANY DIFFERENT SCHOOLS AND CONVENTIONS FROM COAST TO COAST, AS WELL AS ABROAD. IN HIS SPARE TIME, HE WRITES ARTICLES FOR MAGAZINES AND WRITES BOOKS. CURRENTLY, ED WORKS AS A TECHNICAL DIRECTOR AND 3D ARTIST AT SAS STUDIO PRODUCTIONS, A PRODUCTION COMPANY IN RALEIGH, NORTH CAROLINA, USA. HE IS THE PRESIDENT OF THE LOCAL RESEARCH TRIANGLE SIGGRAPH CHAPTER. HE ALSO PRODUCES TRAINING VIDEOS FOR ASPIRING SOFTIMAGE|XSI ARTISTS, RUNS ONE OF THE LARGEST SOFTIMAGE|XSI RESOURCE SITES, AND TEACHES SOFTIMAGE|XSI OVER THE INTERNET VIA MESMER, INC. MORE INFORMATION ABOUT ED HARRISS CAN BE FOUND ON THE WEB AT: WWW.EDHARRISS.COM

Watch "Specidemons" Now!
www.DaveSchool.com

Planning a career in Visual Effects?

It's Time.
Get Serious.
Call DAVE.

DAVE SCHOOL
THE SCHOOL THAT MAKES MOVIES.

Located on the backlot of Universal Studios Orlando
One-Year Program in Animation and Vfx for TV, Film, & Games
Training in LightWave 3D, After Effects, Boujou and MAYA
On Site Green Screen and Motion Capture Stage
Approved for Foreign Students and Veteran Training
Financial Aid is available to those who qualify

The Digital Animation & Visual Effects School, Universal Studios in Orlando Florida (407) 224-3283

HDRI 3D

LIKE TO BE FEATURED IN OUR GALLERY?

Arsenal's FA Cup
Alex Dinnin

Seeing Is In The Mind
Charli Siebert

Surprising moment
Gunther Heinrich

Now accepting submissions at
http://www.hdri3d.com/gallery.htm

What do you think? What do you want?
Tell us at http://www.hdri3d.com/vote

Does a bonus CD come with HDRI 3D?
No, because our readers can download each issue's resources at http://www.hdri3d.com/resources

Moving?
Subscribers, make sure you email your new address to moving@hdri3d.com

XSI base.com
Softimage XSI Community

Forums
Jobs
Interviews
Plugins
Shaders
Contests

www.xsibase.com

BY BERNARD LEBEL

SCRIPTED RENDERING IN SOFTIMAGE|XSI

One of the areas where SOFTIMAGE|XSI shines is rendering. The artist has a large rendering toolset: passes, overrides, satellite rendering, batch rendering, command line rendering, and so on. Now BatchServe ships freely with SOFTIMAGE|XSI, which opens a lot of possibilities for network rendering.

All these features make SOFTIMAGE|XSI a serious rendering solution. However, there is a way to take all of this and bring it further: scripting. Out of the box, the rendering tools are great, but they often require some work from the artist, and lots of repetitive tasks are involved. Used through scripting, rendering can become a real fun experience, where repetitive tasks, boring work, and cumbersome workflow can be minimized.

In this article, we will explore some of the possibilities scripting offers to rendering artists. We will cover many topics. We will not go too deeply into each of them, but instead I will give you the information you need to get the scripts done, like relevant notes about the SOFTIMAGE|XSI object model, system commands, programming tricks, plus many code examples.

This article is intended for users who are somewhat proficient with scripting. It need not be advanced proficiency (this article is not advanced), but being familiar with the object model, parameters, looping techniques, and the like will definitely help. In this article, I will use Python as the scripting language, because this is my main language.

INSTALLING PYTHON

If you are a Windows user, Python is not supplied with Windows; it has to be installed separately. You can download the installer here:

http://www.python.org/download/

You may download Python 2.3.4 or 2.4, but keep in mind that I have not tested the latter so I can't garantee my examples will work perfectly in it. You will download an executable file. Run it, and choose a location for installing the Python directory. You should install it at the root of a drive, simply because it will make paths shorter.

Once this is installed, you will need the pywin32 module: http://prdownloads.sourceforge.net/pywin32/pywin32-203.win32-py2.3.exe

Install this executable; this will make Python an available language in SOFTIMAGE|XSI. You see, SOFTIMAGE|XSI uses active languages for scripting. In short, active languages are languages that can create active objects. Active objects are used to interact with other softwares and the operating system. The pywin32 executable will install the active part of Python, which will make it available in SOFTIMAGE|XSI.

Launch SOFTIMAGE|XSI, and choose Python ActiveX Scripting Engine.

Linux users do not have to worry about this, because Python is supplied with most builds of Linux (at least the ones supported by SOFTIMAGE|XSI), and the active binaries are supplied with SOFTIMAGE|XSI Linux. However, if you are using an old build of Linux (like Fedora Core 1), you may want to upgrade Python to a more recent version. Here I can't help you, as I've never installed anything myself in Linux.

A FEW USEFUL TIPS

Python has its particularities in SOFTIMAGE|XSI. It does not behave exactly as the other langages, so you have to be aware of a few things. The following things can be found in the scripting documentation, but I summarized them for your convenience.

- Use None to specify a null argument.
- True and False require a capital first letter.
- You can use single and double-quotes for strings.
- You have to use the Application class for commands, including SOFTIMAGE|XSI and custom ones. To call a command, use:

```
Application.commandname()
```

You can instantiate the Application class to something shorter, like xsi.

- You also have to use the Application class for things like activesceneroot, selection() and classname().

- To use constants, you need to import them. I recommend you do it this way:

```
from win32com.client import constants as c
```

This will import the constants class and instantiate it as c (much shorter) in one go.

Now when you want to use a constant, you can do it this way:

```
oLights = xsi.activesceneroot.findchildren( ‘’, c.siLightPrimType, None, True )

oLights = xsi.SIFilter( None, c.siLightFilter, True, c.siQuickSearch )

xsi.logmessage( ‘Hello XSI world!’, c.siInfo )
```

Another tip with constants is that once you have imported them, instead of using the class, you can use a string:

```
oLights = xsi.activesceneroot.findchildren( ‘’, ‘siLightPrimType’, None, True )

oLights = xsi.SIFilter( None, ‘siLightFilter’, True, ‘siQuickSearch’ )

xsi.logmessage( ‘Hello XSI world!’, ‘siInfo’ )
```

- To create an XSICollection and other SOFTIMAGE|XSI native ActiveX objects, you need to first import the win32com module, then dispatch the win32com class and create a SOFTIMAGE|XSI factory intrinsic object:

```
import win32com

xsifactory = win32com.client.Dispatch( ‘XSI.Factory’ )

oCollection = xsifactory.CreateObject( ‘XSI.Collection’ )
```

- One of the big problems with Python in SOFTIMAGE|XSI is that most of the time you deal with object pointers, and not objects themselves. So, when you try to access parameter values, you get COM errors and such. To deal with this, make sure to get the object via the scene dictionary:

```
oLights = xsi.SIFilter( None, c.siLightFilter, True, c.siQuickSearch )

for oLight in oLights:

        oLight = xsi.dictionary.getobject( oLight )
```

Note that this technique will not work on object sub-components (vertices, edges, polygons, etc.).

PASSES

To access passes, you have to start traversing the scene hierarchy from higher than the scene root: the active project.

```
# Get passes

oPasses = xsi.activeproject.activescene.passes

xsi.logmessage( ‘I have ‘ + str(oPasses.count) + ‘ passes in the scene.’ )

# Iterate individual passes

for oPass in oPasses:

        xsi.logmessage( ‘Pass name: ‘ + oPass.fullname )
```

Now this is where it becomes tricky. Despite the RenderOptions object being a property, you can't access it as if it was a property. This will fail:

```
oProp = oPass.properties( ‘RenderOptions’ )
```

Instead you have to get the object from the SOFTIMAGE|XSI dictionary:

```
oProp = xsi.dictionary.getobject( oPass.fullname + ‘.RenderOptions’ )
```

Now you can access any parameter from the render options:

```
# Get passes

oPasses = xsi.activeproject.activescene.passes

# Iterate individual passes

for oPass in oPasses:

        xsi.logmessage( ‘Pass name: ‘ + oPass.fullname )

        # Get pass render options

        oProp = xsi.dictionary.getobject( oPass.fullname + ‘.RenderOptions’ )

        # Print end frame value for this pass

        xsi.logmessage( ‘Pass end frame: ‘ + str(oProp.parameters( ‘EndFrame’ ).value) )
```

When you create a new pass, you can either retrieve it from the SOFTIMAGE|XSI dictionary, or you can find it in the pass list of the scene:

```
# Create new pass

xsi.createpass( ‘Pass’, ‘MyNewPass’, None )

oNewPass = xsi.dictionary.getobject( ‘Passes.MyNewPass’ )
oNewPass = xsi.activeproject.activescene.passes( ‘MyNewPass’ )
```

To list all the objects under a pass, you can run a low-level command called EnumElements().

```
# With a pass selected, run enumelements to collect the objects under that pass
oElems = xsi.enumelements( xsi.selection(0) )
# Iterate the objects under the pass
for oElem in oElems:
        xsi.logmessage( oElem )
```

So after that, you will access pass objects like you did for the RenderOptions, via the SOFTIMAGE|XSI dictionary. Let's say you want to run through the Environment shaders. You can use the EnumElements() command on the EnvironmentShadersStack, because there is no way in the object model to access this information:

```
# Get environment shaders stack of the selected pass
oEnvStack = xsi.dictionary.getobject( xsi.selection(0).fullname + ‘.EnvironmentShaderStack’ )
# Get list of shaders from environment shaders stack
oEnvShaders = xsi.enumelements( oEnvStack )
# Iterate list of environment shaders
for oEnvShader in oEnvShaders:
        xsi.logmessage( oEnvShader )
```

PARTITIONS

Now that we know how to handle passes, we can start looking into partitions. Partitions are a special case, because the way they are displayed in the SOFTIMAGE|XSI interface doesn't match the way they are structured in the SOFTIMAGE|XSI SDK. The two background partitions are placed directly under the pass, but the other partitions, despite being listed right under the pass like the background ones, are in fact placed into a partition container, pretty much like environment shaders are placed in the EnvironmentShaderStack container. The difference is that the partition container is not visible in the SOFTIMAGE|XSI interface, which can lead to lots of script errors.

So if we go back to one of our previous examples:

```
# Get selected passes objects
oElems = xsi.enumelements( xsi.selection(0) )
# Iterate objects under pass
for oElem in oElems:
        xsi.logmessage( oElem )
```

This would print:

```
#INFO : Passes.Default_Pass.Partitions
#INFO : Passes.Default_Pass.Background_Objects_Partition
#INFO : Passes.Default_Pass.Background_Lights_Partition
#INFO : Passes.Default_Pass.RenderOptions
#INFO : Camera
#INFO : Passes.Default_Pass.EnvironmentShaderStack
#INFO : Passes.Default_Pass.OutputShaderStack
#INFO : Passes.Default_Pass.VolumeShaderStack
#INFO : Passes.Default_Pass.Name
```

The first object printed, Partitions, is the partition container I mentioned above. To list the partitions it contains, you have to once again use an EnumElements() command on it:

```
# Get selected passis objects
oElems = xsi.enumelements( xsi.selection(0) )
# Iterate pass objects
for oElem in oElems:
        # Check if pass object has ‘Partitions’ in its name,
        # which means it would be the custom partitions container.
        if ‘Partitions’ in str(oElem) and not ‘Background’ in str(oElem):
                # Get the partition container elements (that is, custom partitions)
                oCustomPartitions = xsi.enumelements( oElem )
                # Iterate partitions
                for oPartition in oCustomPartitions:
                        xsi.logmessage( oPartition.fullname )
                        oPartition = xsi.dictionary.getobject( oPartition )
```

At last, you need to get the partition objects via the SOFTIMAGE|XSI dictionary.

Once you have gotten to the partitions, you can start either working on them or their members. In the SOFTIMAGE|XSI object model, partitions are groups; they share the same properties and methods. So, for example, if you want to print the number of objects a partition has, you can simply print the count property of the partition members property.

```
xsi.logmessage( oPartition.members.count )
```

You can add a material to the partition with the addmaterial() method. You can also add an override with the addproperty('Override') method.

To move objects from one partition to another, you can either use the *CopyPaste()* or *MoveToPartition()* commands. Unfortunately, the *addmember()* method, which works for groups, does not not seem to work with partitions.

So far, we have seen how to move downward the object model to get to partitions, but what if we wanted to move upward? Let's say we have an object and we want to know not only the partition it is a member of, but also what pass this partition is in.

To get the partition, we have to evaluate the owners of the object. This will return a collection of at least two group objects; one is the layer, the other a partition. In the SOFTIMAGE|XSI SDK, groups, layers and partitions are all groups, with very subtle distinctions. So, to get to the partitions owning an object, you have to filter out the layers, and then check if the owner of the group is a pass, which means it's a partition. This is a lot of work, but it also allows you to get to the pass object.

```
# Iterate owners of selected object

for oOwner in xsi.selection(0).owners:

        # Check if owner is a group, but not a layer

        if oOwner.type == c.siGroup and not xsi.classname( oOwner ) == 'Layer':

                # Iterate owners of the group

                for oGroupOwner in oOwner.owners:

                        # Check if group owner is a pass

                        if oGroupOwner.type == '#Pass':

                                xsi.logmessage( 'Partition "' + oOwner.name + '" belongs to pass "' + oGroupOwner.name + '".' )
```

MATERIALS

There isn't much to say about materials, as they're generally pretty straightforward to script. Each geometric and hair object has a material property, which makes it easy to access them.

However, keep in mind that scripting-wise, a material is also a property of an object and can be accessed like any other property. So, as soon as you add a material to any object, including groups and partitions, that object also receives an additional property object.

```
# Iterate local properties of selected object

for oProp in xsi.selection(0).localproperties:

        # Check if local property is a material

        if oProp.type == c.siMaterialType:

                xsi.logmessage( oProp )
```

When looping over the properties of an object, if you are looking for materials, it is very important that you iterate the local properties rather than the properties. The problem is that an object can receive two materials simultanously: one is local, the other is applied (inherited). If you print the properties of the object, only the local one will be printed. However, if you print the local properties, both of them will be printed.

NOTE: you can verify this by selecting an object, and apply a material. Then branch-select the object, and apply another material. Now, toggle the scope of the Explorer between Local Properties and Applied Properties, and notice that one of them is not visible when the scope is set to Applied Properties. Then run this code to further observe the problem:

```
for oProp in xsi.selection(0).properties:

        if oProp.type == c.siMaterialType:

                xsi.logmessage( oProp )

xsi.logmessage( '----------------------------------------' )

for oLocalProp in xsi.selection(0).localproperties:

        if oLocalProp.type == c.siMaterialType:

                xsi.logmessage( oLocalProp )
```

Finally, in SOFTIMAGE|XSI 4.0 and later, materials are no longer exclusive properties of objects. Each material is an item of a material library. To know the library owning the material, simply use the library property on the material:

```
# Print the library of the selected object's material

xsi.logmessage( xsi.selection(0).material.library )
```

SHADERS

As straightforward as scripting materials can be, scripting shaders is all but straightforward. Shaders is an area where the user is pretty much left to himself and has to use undocumented commands and messy loops to get the result.

First, you can retrieve any shader in the scene very easily, with a command not documented.

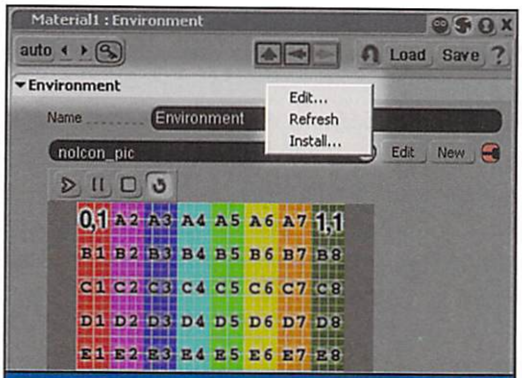
```
# Get change_range shaders, returns a collection

oShaders = xsi.findobjects( None, "{BDE291C1-077A-11D2-8A1A-00A0C9892542}" )

# Check if shaders were found

if oShaders.count > 0:

        # Iterate found shaders

        for oShader in oShaders:

                xsi.logmessage( oShader )
```

This LogMessage() command will print not only the name of the shader, but also the path of the shader. If the shader is in a material, the full path to the shader will get printed. If the shader is among the environment shaders of the pass, the pass is printed.

Now what is that FindObjects() command all about? That big set of characters and numbers is the ClassID of the shader. The ClassID is a unique number that identifies the class of shader you are dealing with. Each shader has its unique class, so by using that class identifier, you can retrieve all instances of that shader in the scene. To find out the ClassID of a shader, create the shader in the Render Tree (or any other place where you can access the property page of a shader), open the property page, hold down Ctrl and Shift, then right-click on the shader name that is next to an arrow (black text on white background). A little menu shows up, choose Edit...(See Image Above Right)

The third line of the text file that opens, that starts with Reference, holds the ClassID of the shader. Use that in your script to retrieve these shaders.

Another way to retrieve shaders is to traverse the Render Tree of materials. This will be a lot more accurate than using the FindObjects() command, because you can include search criterias in your code. However, traversing the Render Tree is a lot of work. Those familiar with Python will greatly miss the os.path.walk() method for traversing directory structures!



RIGHT-CLICK WITH THE CTRL AND SHIFT KEYS HELD ONTO THE TITLE OF THE SHADER, NEXT TO THE ARROW, AND CHOOSE EDIT...TO GET THE CLASSID OF THE SHADER

There are two ways to get the shaders connected into a shader. One is to use the shaders property of the shader. If the collection it returns has a count higher than 0, then the shader has other shaders connected. The other way is to loop over all the parameters (inputs), to test if the parameter has a source. When the parameter has a source, it means that something might be connected, but can also mean there is an expression or a linked parameter. So, after that, you have to get the source itself and check if it is a shader.

Here I will present a technique to traverse the Render Tree, then I will show you the code.

First, I start by creating a collection. This collection is used to store individual shaders of a material. This collection is set to accept the same item only once. Then the material, which is a shader, is added to the collection, and will be the starting point of the Render Tree traversal.

The actual traversal of the Render Tree then starts with a while loop that will run as long as new shaders are found during an iteration. The current collection count is recorded, then all the shaders already in the collection are looked up. As it has been described earlier, each parameter of the shaders is evaluated. If shaders are connected, these shaders are added to the collection.

When all shaders of the collection have been tested, the collection count is tested. If it has not changed from when it was recorded (at the beginning of the while loop), it means that no new shaders were found. Considering that all shaders of the collection have been tested, it means there are no more shaders to be found, so all shaders of the Render Tree have been collected and the while loop stops there. Otherwise, if the collection count has increased, new shaders have to be tested, and the while loop starts over again.

```
# Create empty collection to store shaders for each material

oShaders = xsifactory.CreateObject( 'XSI.Collection' )

oShaders.unique = True

# Set default collection count value

iShaderCount = 0

# Add material shader to collection

oShaders.add( oMat )

bLoop = True

# Start loop that evaluates each shader of the collection

while bLoop == True:

    # Record the collection count

    iShaderCount = oShaders.count

    # Iterate collection of shaders

    for oShader in oShaders:

        # Iterate parameters of shader

        for oParameter in oShader.parameters:

            # Check if shader input has source (something connected)

            if not oParameter.source:

                pass

            else:

                # Add source shader to shader collection

                oShaders.add( oParameter.source )

    # Check if shaders were collected, compare current collection count to recorded collection count

    if oShaders.count > iShaderCount:

        # Maintain loop condition and start another round of iteration

        bLoop = True

    else:

        # If no shaders were collected, stop loop

        bLoop = False
```

An advantage of this script is that you can insert search criterias once the collection of shaders iteration has started. For example, if I wanted to get all the Image nodes, I would create an empty collection (also set to uniqueness), then test the collected shaders to see if they are image nodes (by getting their ClassID), and add them to that collection. Below is a short version of the previous script example.

```
# Create tool that provides access to shader ClassIDs

oDatabase = XSIUtils.DataRepository

# ..... jump to loop

        for oShader in oShaders:

            # Get shader ClassID

            sShaderID = oDatabase.getidentifier( oShader, 3 )

            # Compare the ClassID we got to the Image shader one

            if sShaderID == sImageShaderID:

                oImageShaders.add( oShader )
```

Now that you have all the Image shaders of the material, you could loop over them to change parameters or get their clip.

XSIBATCH & RAY3

XSIBATCH is an important part of the rendering. While in SOFTIMAGE|XSI, most will use only the render region tool, and then make the final rendering using XSIBATCH. However, using scripting, a lot more than rendering can be done with XSIBATCH. You can, in fact, open a scene and edit it entirely through scripting. I find this very handy when I have to deal with very large scenes, because it uses very little memory and I don't have to wait for the GUI to refresh.

For example, I once had a scene that was so large that it would crash while exporting to mi2. So I launched an XSIBATCH command prompt, then ran a .bat file that had a command to execute a script. The script had two commands: OpenScene() and ExportMI2File(). Unfortunately, it also crashed, but it shows that with scripts you can still interact with the scene without the interface.

To interact this way with SOFTIMAGE|XSI, you'll have to write down two files. The first file is a batch file (a text file with a .bat extension), and you'll need to write a script file execution instruction. The second file is the script itself, with all the commands you want to be run. You can also write any script you want, it may not just be commands.

Another use of scripting with rendering is to edit mi2 files. There are several features of mental ray that you can take advantage of only with mi2 file rendering, like user-defined area lights, large BSP acceleration, light maps, and so on. So instead of opening huge mi2 files in a text editor, you might want to write scripts that would look for specific text elements in the file and edit them, or add informations in certain areas. Regular expressions would definitely be the way to go.

SYSTEM COMMANDS

Finally, let's briefly discuss system commands. System commands consist of commands that you send from a script to an application. This command is an instruction that the target application can recognize, and arguments are provided with the instruction so the task can be completed. System commands are a powerful tool to handle tasks that require interaction between several applications, operating systems, and computers across a network.

One use that I will discuss here is to fasten the conversion of .pic files to .map files. Basically, with a script, you can tell mental ray to perform a conversion in a directory of your choice.

First you need to import the os module. This is a big Python module that gives a lot of functionality to handle files and directories, and it is with this module that you send system commands. Then you need to change the current directory, like you would do in a command prompt. Point to the appropriate location of the *imf_copy.exe* application. When you send the system command, the script will look for the application in the current working directory, no other.

Then, specify the directory where you want the conversion to take place. After that, all you have to do is build the command that will tell imf_copy to generate the .map file:

```
import os
# Change current working directory to point to imf_copy (be careful to choose the good cpu)
os.chdir( ‘C:\\Softimage\\XSI_4.2\\Application\\bin\\nt-x86-p4’ )
# Define path where imf_copy will input the .pic file and output the .map file
sPath = ‘C:\\Temp’
# Specify file name to convert and file name to output
sInput = ‘myTexture.pic’
sOutput = ‘myTexture.map’
# Build command to send to imf_copy
# The command is identical to what you write in a SOFTIMAGE|XSI shell to perform the conversion
sCommand = ‘imf_copy ’ + sPath + ‘\\’ + sInput + ‘ ’ + sPath + ‘\\’ + sOutput
# Send system command to imf_copy
os.system( sCommand )
```

The problem with this command is that it will always expect the same file to convert. This might not be ideal if you're working on several textures at a time. What you could do is set up a script that checks all the files in the directory and compare the modification dates of both the .pic and .map files. If it finds a .pic newer than the .map version, it performs a conversion.

So what we will do here is to list the content of the target directory, check all files with the same name before any extension and padding, and then check if the .pic version is newer than the .map version. We will need another module called string, whose purpose is to handle...strings! The next piece of code is heavily documented, so read those comments to easily follow the steps:

```
import os
import string
def runCommand( sPath, sPicFile, sMapFile ):
    “”” This function simply composes the final command to run,
    using the path and file names provided.
    Then it executes the command. “””
    # Build command
    sCommand = ‘imf_copy ’ + sPath + ‘\\’ + sPicFile + ‘ ’ + sPath + ‘\\’ + sMapFile
    # Send system command to imf_copy
    os.system( sCommand )
# Change current working directory to point to imf_copy (be careful to choose the good cpu)
os.chdir( ‘C:\\Softimage\\XSI_4.2\\Application\\bin\\nt-x86-p4’ )
# Define path where imf_copy will input the .pic file and output the .map file
sPath = ‘C:\\Temp’
# Get directory entries (both files and directories)
aEntries = os.listdir( sPath )
# Iterate directory entries
```

```
for sEntry in aEntries:

    # Split file name into a list. This will seperate the various file name elements, using the dots in the name.

    aFile = string.split( sEntry, ‘.’ )

    # Check the created list to see if the entry has a pic extention. Logically, the extension would be the last element in the list.

    if aFile[-1] == ‘pic’:

        sPicFile = sEntry

        # We have to compose a fictive file name with a .map extention, then check if that file exists among the directory entries.

        # But we have to determine if the file has a padding. To do this, check the length of the list.

        if len( aFile ) == 2:

            # File has no padding, simply add the custom extention to its name

            sMapFile = aFile[0] + ‘.map’

        elif len( aFile ) == 3:

            # File has padding, so get the padding from the list and insert it between the file name and the extension

            sMapFile = aFile[0] + ‘.’ + aFile[1] + ‘.map’

        # Next steps are to check if the file name we composed actually exists in the directory entries

        # Here we have a pic file that does not have a .map version

        if sMapFile not in aEntries:

            # Call function that runs system command

            runCommand( sPath, sEntry, sMapFile )

        # Here we have a pic file that does have a .map version, so we'll need to compare modifcation dates.

        else:

            # Get last modification time

            iPicTime = os.path.getmtime( sPath + ‘\\’ + sPicFile )

            iMapTime = os.path.getmtime( sPath + ‘\\’ + sMapFile )

            # Compare modification dates

            if iMapTime < iPicTime:

                # Call function that runs system command

                runCommand( sPath, sPicFile, sMapFile )
```

CONCLUSION

Covering all aspects of rendering with scripting would have required me to write a book. For example, we have not seen the scripting of lights, textures, the FX Tree, reference models, text file input/output, image clips and image sources, overrides, render farms (that alone would make most of the book!), BatchServe, plus the countless other things related to rendering. Perhaps that could be material for a future article … do not hesitate to send letters to the editor of HDRI 3D if you want it so!

Still, I hope this short article gave you some interesting scripting ideas. Personally, I find that almost anything related to rendering benefits from scripted tools, since it often involves many tedious and repetive tasks. For example, disabling the reflectivity on hundreds of materials that are all different can be time consuming, especially if you have to do it many times. This can be shortened dramatically by using the Render Tree traversal code I provided to iterate every shader in the scene. Plus, I find scripting a fun challenge of logic and engineering.

On a last note, I invite you to check out my personal Web site's script page, where you will find lots of rendering-related scripts, in Python. Happy scripting!

RELEVANT LINKS:

BERNARD LEBEL HOME PAGE: http://www.bernardlebel.com

BERNARD LEBEL SCRIPT PAGE: http://www.bernardlebel.com/scripts.htm

PYTHON DOWNLOAD: http://www.python.org/download/

PYWIN32 MODULE: http://prdownloads.sourceforge.net/pywin32/pywin32-203.win32-py2.3.exe

SOFTIMAGE|XSI FORUMS AND RESSOURCES: http://www.xsibase.com

SOFTIMAGE|XSI RESSOURCES: http://www.edharriss.com

SOFTIMAGE HOME PAGE: http://www.softimage.com

BERNARD LEBEL AFTER A FEW YEARS IN GRAPHIC DESIGN, BERNARD MADE THE MOVE TO 3D ANIMATION. HE HAS BEEN WORKING IN 3D GRAPHICS FOR OVER FOUR YEARS. LIGHTING AND RENDERING ARE HIS FAVORED AREAS. HE WORKED IN VARIOUS STUDIOS LIKE *MULITVET MEDIA, TWIST!IMAGE, CINÉGROUPE, ACTION SYNTHÈSE, METEOR STUDIOS*. HE HAS BEEN DOING SCRIPTING SINCE SPRING 2003, AND PYTHON SCRIPTING SINCE JUNE 2004. HE HAS BEEN USING SOFTIMAGE|XSI SINCE DECEMBER 2001. HE ALSO CREATED SOFTIMAGE|XSI TRAINING MATERIAL FOR *3DTUTORIAL.COM*, AND IS WORKING ON A SECOND SET.

BY DEUCE BENNETT

The Graph Editor is Your Best Friend!

For those of us who have been working with computer graphics since the dark ages, we remember the painful difficulties of trying to create fluid movements of camera, objects, and characters. We didn't *have* a Graph Editor. All adjustments had to be made by seat-of the-pants keyframe adjustments. We had no way of determining how a change would affect the flow of movement until we created another preview. Making alterations such as slowing a move from full speed to a stop without a jerk became a frustrating ordeal. Only the very dedicated geeks prevailed through these primitive early developments.

While we were growing into full-fledged animators, the software was also evolving, and still is. Among the most valuable tools we gained to make animators' lives easier was a device to help facilitate the perfection of movements without requiring sacrificing large portions of your day. We got a Graph Editor. Then, that was improved upon when we no longer had to separate channels of animation with a string of nulls. We could animate and control independent channels in the Graph Editor. Oh, the happiness!

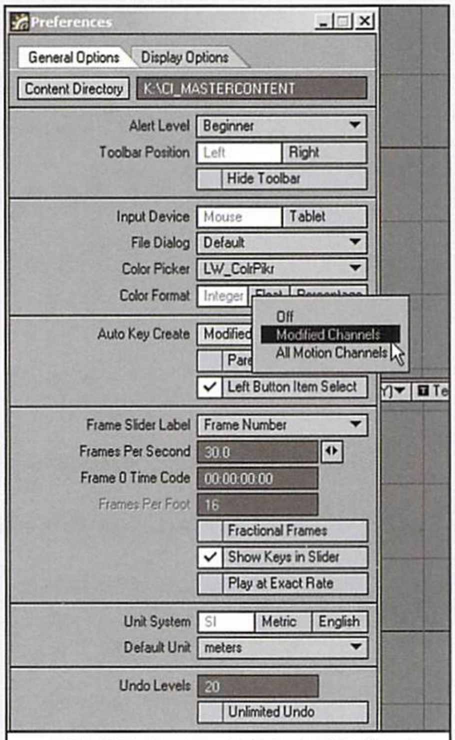
Amazingly, many Wavers who enjoy all the other tools in their favorite software still don't know how to tap the power of this great magic wand. It requires a little understanding to learn how to 'read' those curves and be able to spot rough spots that need first-aid. Also, it requires a chant, a mantra, a 'way of working' to really make smooth animations.

The fewer the keyframes, the smoother the animation and the easier it is to control.

What many new users don't really understand is how to 'read' the curves for a smooth path, or when to use what kinds of curves. Also, many users don't know what controls you have in the Graph Editor that you don't have anywhere else. That's what I hope to impart to you in this article.

Animation, moving things around, in LightWave is a straightforward affair. You set the keyframes, the computer does the in-betweens. Simple. What can separate two animators is the smoothness of moves, or the amazing timing that some skilled artists develop. One tool that has become indispensable, is the graph editor or curve editor.

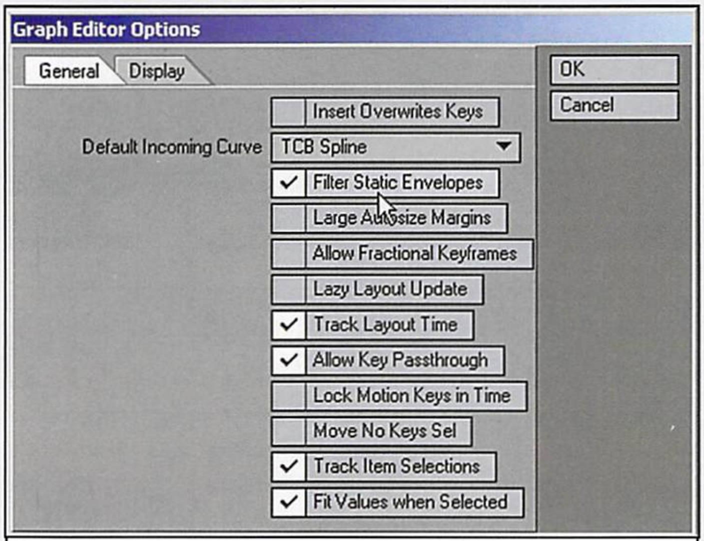
Back in the old days, you couldn't set keyframes for individual channels at different times. At that time, you needed to parent anything that you wanted to make smooth movements with to several nulls – usually one null for each channel that you wanted to separate. This made for some interesting scenes, having to keep all the nulls under control. Thankfully, NewTek at the version just after 5.6, LightWave gained the ability to separate all the channels, and the null was free again! OK, but seriously, the point of this is – if you don't need a keyframe on a channel, don't put one. Remember the mantra. This is the primary problem that many people who do work in "autokey" mode run into when trying to make smooth movements. Make sure, if you are one of those who like to use autokey, that you enable "Modified Channels" in the General Options tab of the Preferences. This way, whatever channel you modify, that will be the only channel you automatically keyframe.



Content Directory: K:\CI_MASTERCONTENT
Alert Level: Beginner
Toolbar Position: Left | Right
☐ Hide Toolbar
Input Device: Mouse | Tablet
File Dialog: Default
Color Picker: LW_ColrPikr
Color Format: Integer
Auto Key Create: Modified
☐ Pare
☑ Left Button Item Select
Frame Slider Label: Frame Number
Frames Per Second: 30.0
Frame 0 Time Code: 00:00:00:00
Frames Per Foot: 16
☐ Fractional Frames
☑ Show Keys in Slider
☐ Play at Exact Rate
Unit System: SI | Metric | English
Default Unit: meters
Undo Levels: 20
☐ Unlimited Undo

99.9% OF THE TIME – MODIFIED CHANNELS

OK, let's move to the Graph Editor. First off, let me mention one thing that you will need to keep handy. That is the option for 'automatically filter static envelopes'. What this does is, if you select an object, light, bone– and it has some envelopes (channels) that only have one keyframe (making it not move), then that channel will not show up in the Graph Editor. This is very handy when you have several objects that you are working with, but initially, it can be frustrating as you might not have any keys set for something that you are just starting with.



Graph Editor Options

General | Display

OK

Cancel

☐ Insert Overwrites Keys

Default Incoming Curve: TCB Spline

☑ Filter Static Envelopes

☐ Large Autosize Margins

☐ Allow Fractional Keyframes

☐ Lazy Layout Update

☑ Track Layout Time

☑ Allow Key Passthrough

☐ Lock Motion Keys in Time

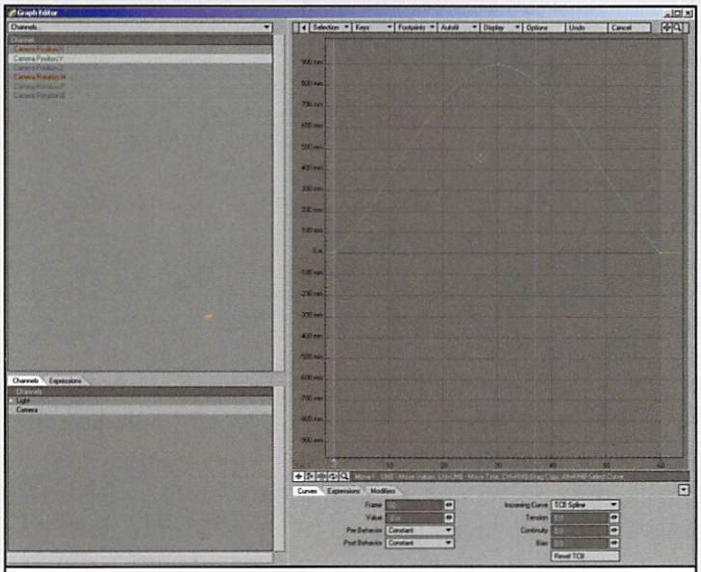
☐ Move No Keys Sel

☑ Track Item Selections

☑ Fit Values when Selected

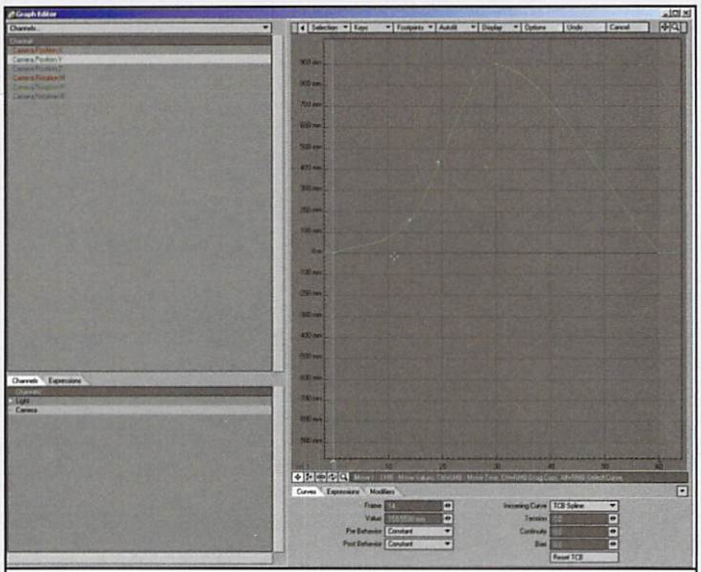
DON'T SHOW ME SOMETHING THAT ISN'T MOVING

OK, here is a very simple graph, total of three keyframes; this is very, very easy to make smooth. A good reason why is because three keyframes, honestly, by their very nature are smooth. In a way, they are like a 3-point polygon and cannot be non-planar, I suppose.



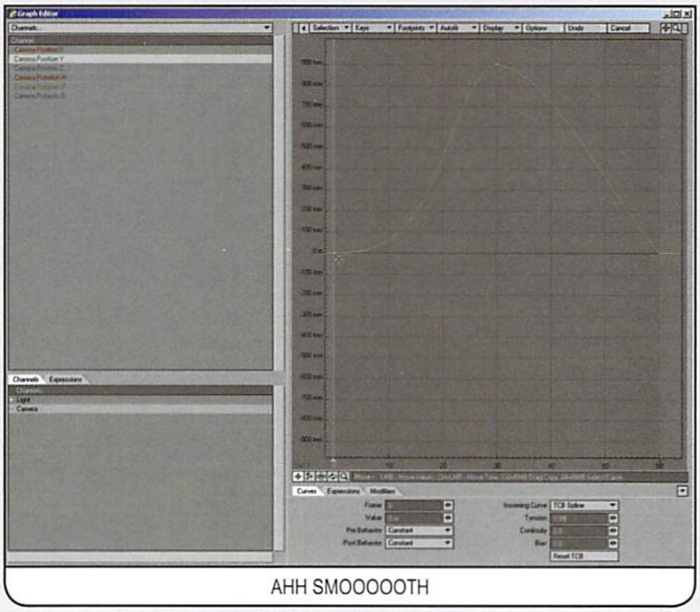
BELL CURVE - REMEMBER BLOWING THOSE IN SCHOOL

There are many times, however, that you have to use more than three.



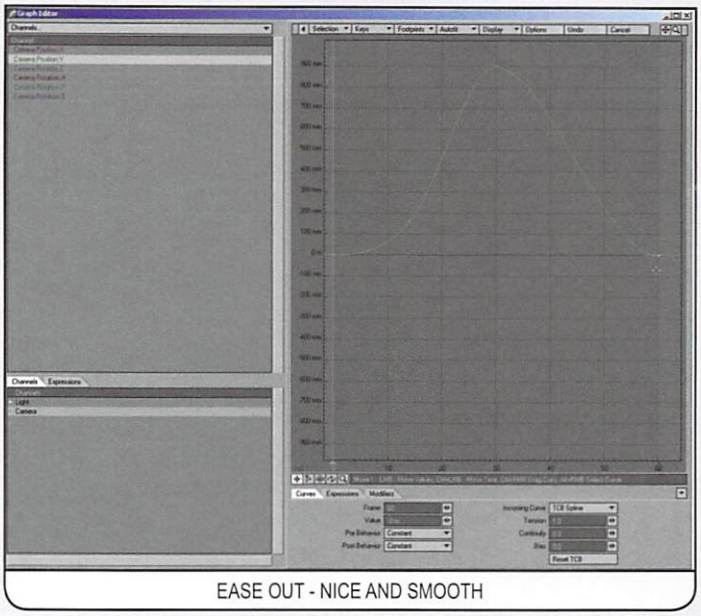
BROKEN SMOOTH

Now, this looks fine, but to me, there is one bad area. That is between the first and second keyframes. Here's where using the TCB controls can smooth out a path.



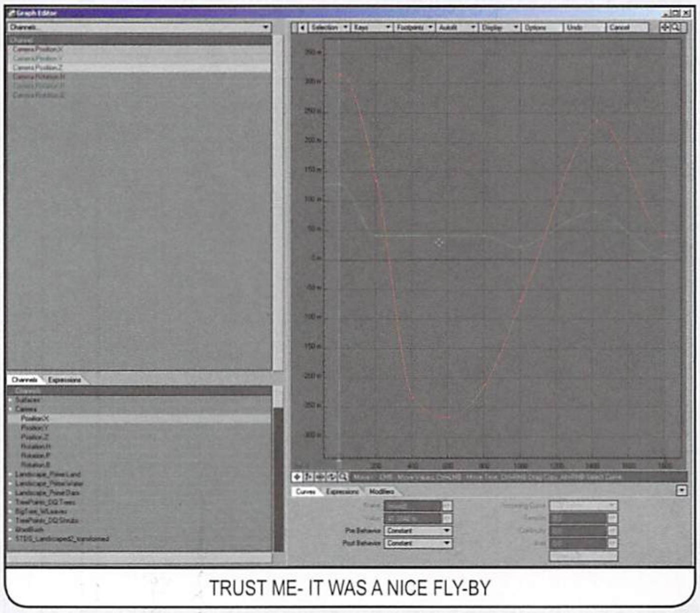
AHH SMOOOOOTH

Just a little sliding and we have a nice ease to that keyframe; adding a Tension of 1 to the last keyframe makes it all the nicer. See *Ease Out* top of next page.



EASE OUT - NICE AND SMOOTH

Here is a camera fly-by graph that was a little more complex. All I typically do is set my keyframes for "where I want to be at what time" – then smooth out the rough spots.

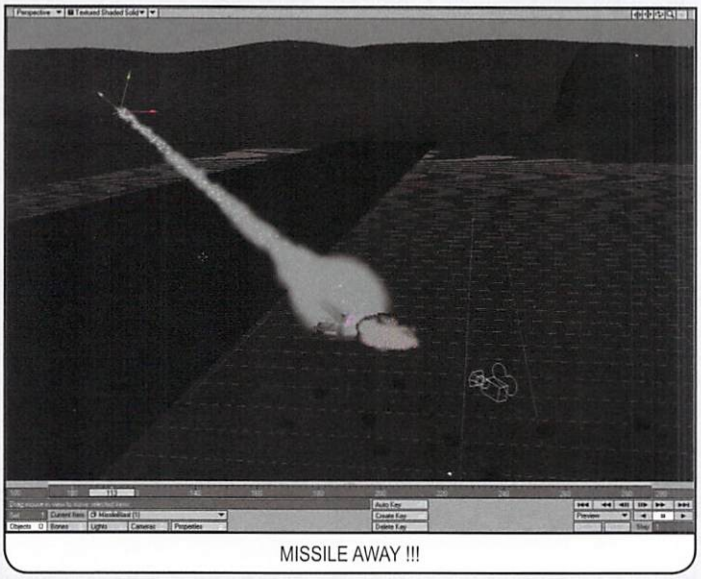


TRUST ME- IT WAS A NICE FLY-BY

This turned out to be a very nice animation.

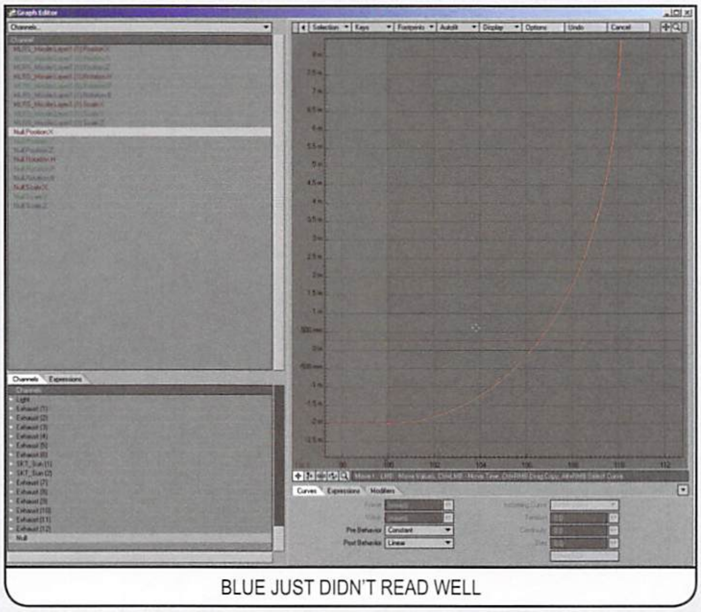
Now, another question that comes up from time to time is, "when do I use the other types of curves?" Well, I'm an old-timer (oh gosh, now it's in print!), so TCB is what I typically use. However, that being said, there are some absolutely good reasons for the other options. Otherwise, why would they be there?

Here's a good example that I run into a whole bunch doing military visualizations. The launching of a missile. PIC_0007.TGA



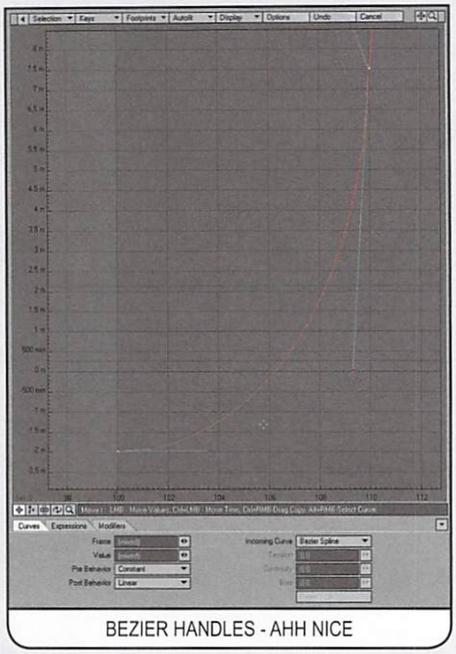
MISSILE AWAY !!!

(Note: the Z channel was what was keyframed, but I copied it to a null so that I could put it on the X channel for clarity) The motion of a missile, slow to start and rapidly, very rapidly, accelerating, is just hard to get, if not possible to achieve, with TCB curves. Remember our mantra – the fewer the keyframes, the smoother the animation and the easier it is to control. So here, my entire missile launch is two keyframes.



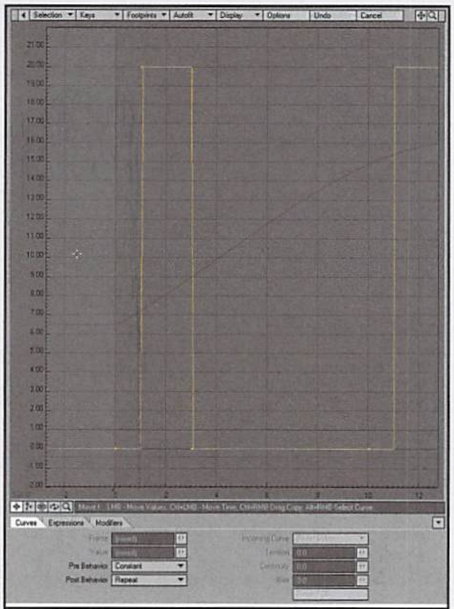
BLUE JUST DIDN'T READ WELL

The 'trick' is that they are Bezier keyframes. See *Bexier Handles* top of next page. With the Bezier handles, it's far easier to achieve that very slow speedup, ending with an acceleration curve. Bezier have handles that you can really pull the curve in with. Hermite don't have this, but Hermite are good for smooth curves. Also, by setting the "Post Behavior" to Linear, the acceleration that I set up continues forever.


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
Curves
Expressions
Modifiers
Frame
Value
Pre Behavior
Constant
Post Behavior
Linear
Incoming Curve
Bezier Spline
Tension
Continuity
Bias

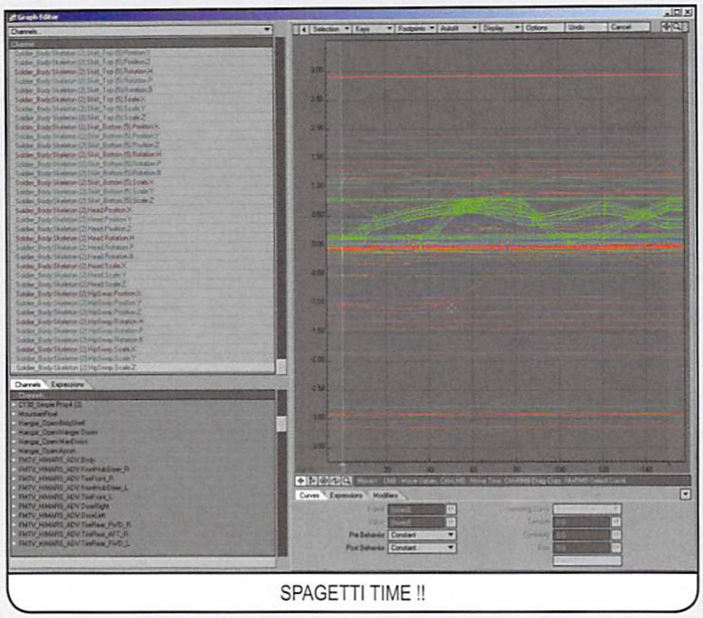
BEZIER HANDLES - AHH NICE

Stepped keyframes; above. this keeps our mantra going with only needing two keyframes for an "on-off" type behavior. You know we used to need three keyframes for this?


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
Curves
Expressions
Modifiers
Pre Behavior
Constant
Post Behavior
Repeat

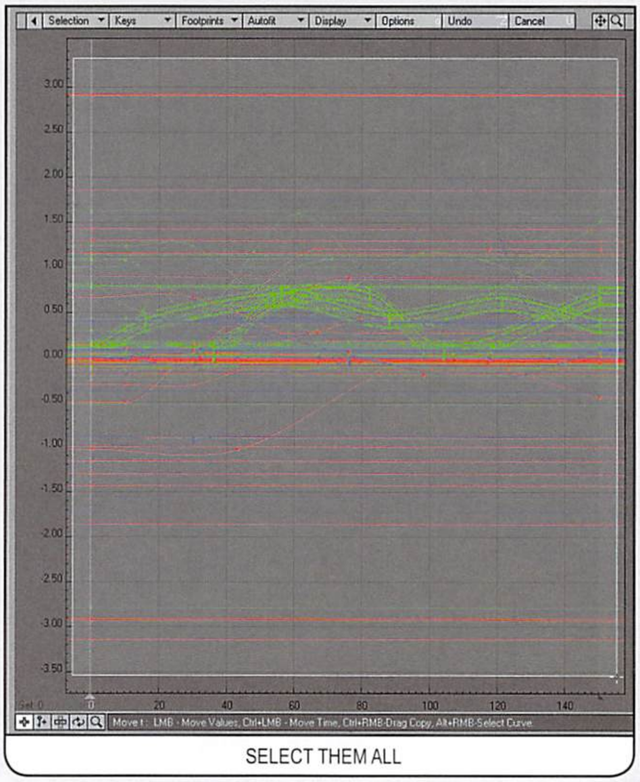
TIME KEEPS ON STEPPIN' INTO THE FUTURE

OK – I'm going to finish this off with a little "Stupid Graph Editor Tricks." I am a HUGE believer in the credo that computers are a labor saving device. Yes, I know, I hear you all – trust me, you use them right? They are. For example, let's say you have an animation that you have already done and are happy with, but in the next scene, you need to have that same motion reversed.

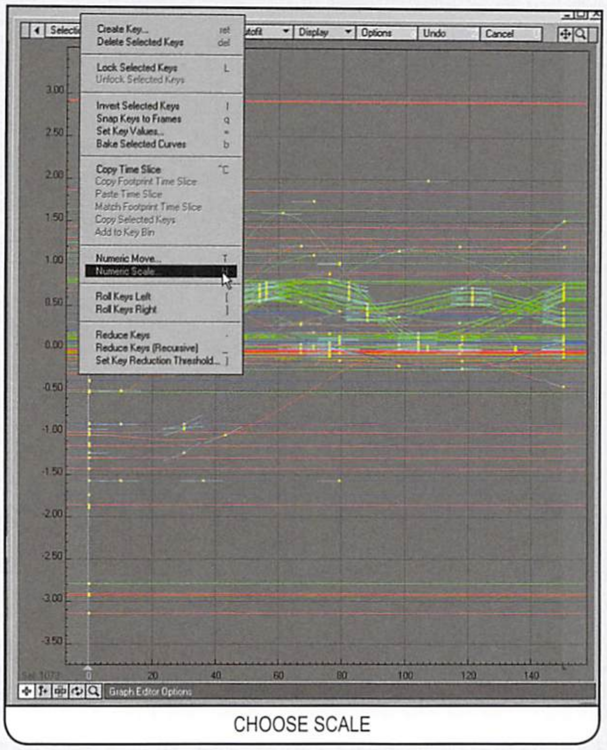

Graph Editor
Channels
Expressions
Curves
Modifiers
Pre Behavior
Constant
Post Behavior
Constant

SPAGETTI TIME !!

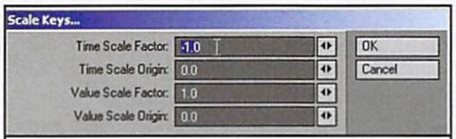
You can't do this in the Scene Editor, because it doesn't understand negative frames with mathematic offsets. Just select all the keyframes of your animation, and choose "Numeric Scale" from the "Keys" menu (See *Choose Scale* on next page) at the top. (See *Numeric Scale* on next page) Type in "minus one" (-1) for your "Time Scale Factor" and you have reversed your animation. (See *Flip It* on next page) Then just move the keys (they are


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
3.00
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
-2.00
-2.50
-3.00
-3.50
0
20
40
60
80
100
120
140
Move t : LMB - Move Values, Ctrl+LMB - Move Time, Ctrl+RMB-Drag Copy, Alt+RMB-Select Curve

SELECT THEM ALL


Create Key...
Delete Selected Keys
Lock Selected Keys
Unlock Selected Keys
Invert Selected Keys
Snap Keys to Frames
Set Key Values...
Bake Selected Curves
Copy Time Slice
Copy Footprint Time Slice
Paste Time Slice
Match Footprint Time Slice
Copy Selected Keys
Add to Key Bin
Numeric Move...
Numeric Scale...
Roll Keys Left
Roll Keys Right
Reduce Keys
Reduce Keys (Recursive)
Set Key Reduction Threshold...
Graph Editor Options

CHOOSE SCALE



Scale Keys...

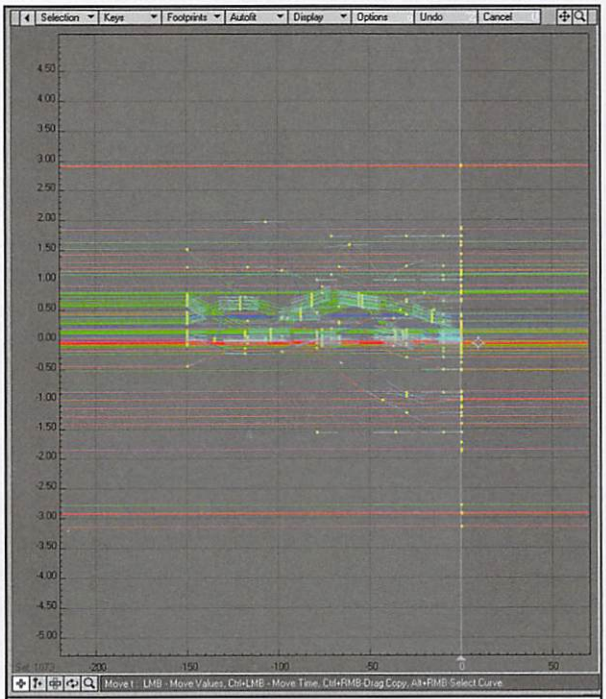
Time Scale Factor: -1.0

Time Scale Origin: 0.0

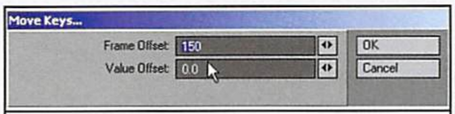
Value Scale Factor: 1.0

Value Scale Origin: 0.0

NUMERIC SCALE OF -1


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
Move t : LMB - Move Values, Ctrl+LMB - Move Time, Ctrl+RMB Drag Copy, Alt+RMB Select Curve

FLIP IT !!

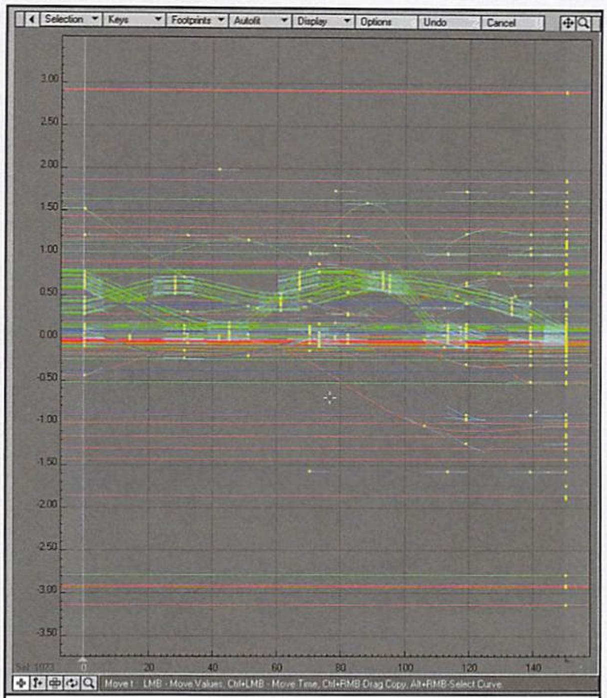


Move Keys...

Frame Offset: 150

Value Offset: 0.0

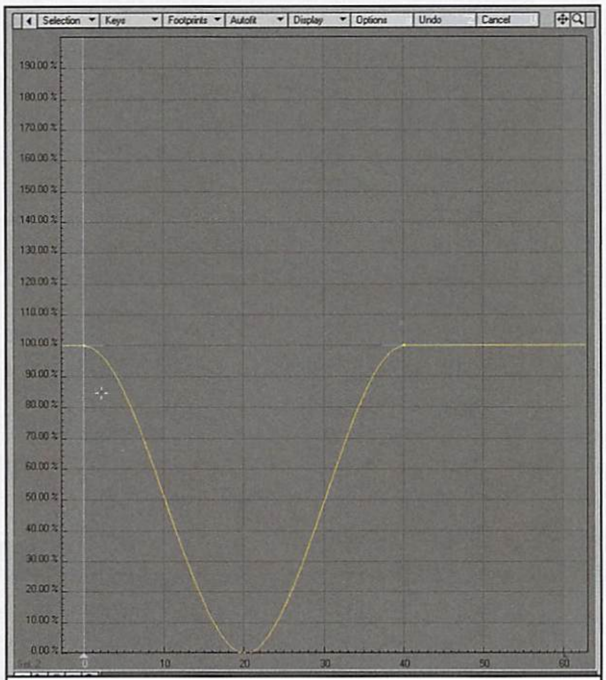
MOVE THEM FORWARD


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
Move t : LMB - Move Values, Ctrl+LMB - Move Time, Ctrl+RMB Drag Copy, Alt+RMB Select Curve

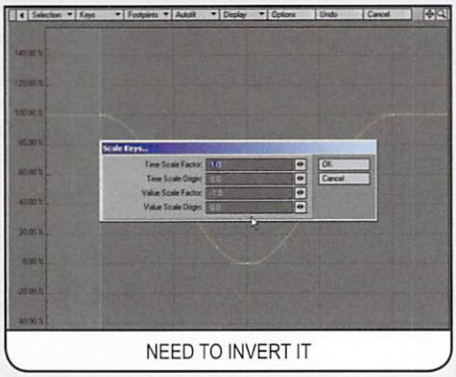
BACK IN THE POSITIVE

still selected) forward the length of your animation. (Bottom left) Mine happened to be 150 frames. (Above)

OK – one last time saver. I do a bunch of muzzle flashes. I typically do them with a light (point light) for a flare and the illumination around the gun muzzle, and an object that I have made that is shaped somewhat "muzzle flash-like" – but the trick is, when the Lens Flare is at 100% brightness, I want my Object to be 0% dissolved. Rather than use an expression, I just copy and paste, then use math to finish it.


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
190.00 %
180.00 %
170.00 %
160.00 %
150.00 %
140.00 %
130.00 %
120.00 %
110.00 %
100.00 %
90.00 %
80.00 %
70.00 %
60.00 %
50.00 %
40.00 %
30.00 %
20.00 %
10.00 %
0.00 %
0
10
20
30
40
50
60

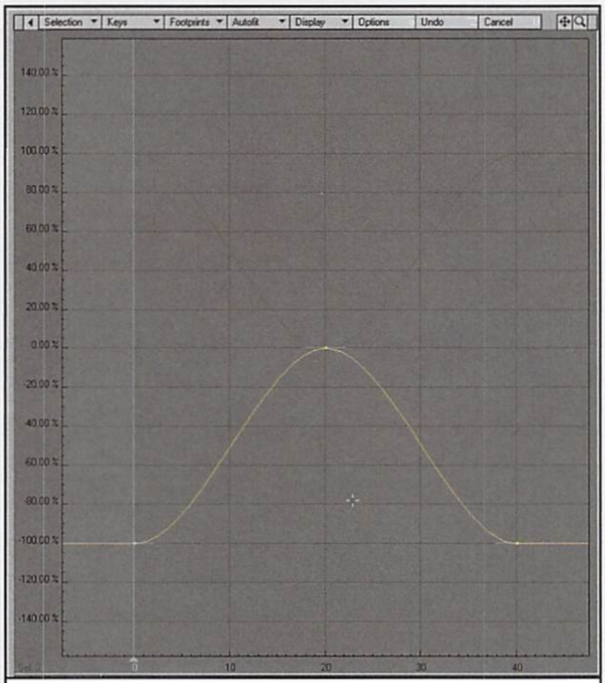
ORIGINAL


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
Scale Keys...
OK
Cancel

Time Scale Factor: 1.0
Time Scale Origin: 0.0
Value Scale Factor: -1.0
Value Scale Origin: 0.0

NEED TO INVERT IT

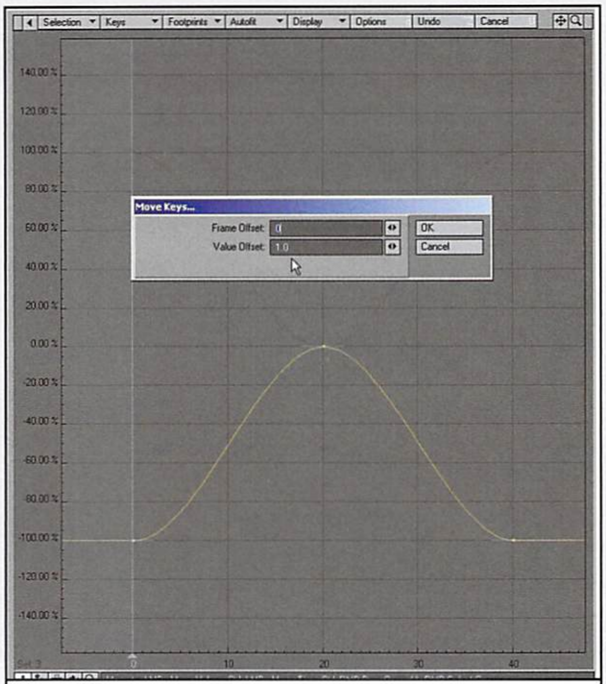
Select all keyframes – select "Numeric Scale" from the "Keys" menu. Enter "minus one" (-1) for "Value Scale Factor" – this 'inverts' the animation, but makes it travel from -100% to 0% and back.


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
140.00 %
120.00 %
100.00 %
80.00 %
60.00 %
40.00 %
20.00 %
0.00 %
-20.00 %
-40.00 %
-60.00 %
-80.00 %
-100.00 %
-120.00 %
-140.00 %
0
10
20
30
40

INVERT

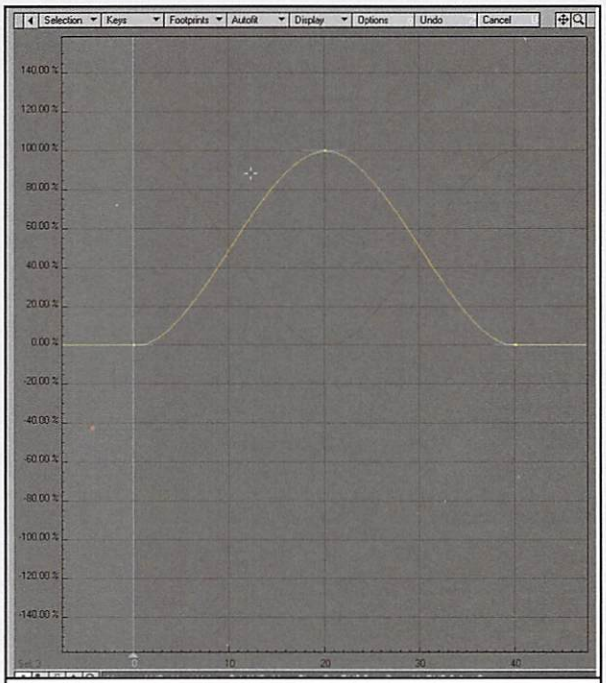
So, choose "Numeric Move" from the "Keys" menu and enter 1.0 for the value offset. There you go, two steps and you can invert an entire curve.

That's all for this issue. Keep it true and I'll see you all around!!


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
Move Keys...
OK
Cancel
140.00 %
120.00 %
100.00 %
80.00 %
60.00 %
40.00 %
20.00 %
0.00 %
-20.00 %
-40.00 %
-60.00 %
-80.00 %
-100.00 %
-120.00 %
-140.00 %
0
10
20
30
40

Frame Offset: 0
Value Offset: 1.0

OFFSET


Selection
Keys
Footprints
Autofit
Display
Options
Undo
Cancel
140.00 %
120.00 %
100.00 %
80.00 %
60.00 %
40.00 %
20.00 %
0.00 %
-20.00 %
-40.00 %
-60.00 %
-80.00 %
-100.00 %
-120.00 %
-140.00 %
0
10
20
30
40

MOVE IT BACK UP

JACK "DEUCE" BENNETT II IS A FREELANCE CGI ARTIST WITH A BACKGROUND IN PHYSICAL SPECIAL EFFECTS FOR MOTION PICTURES AND TELEVISION. DEUCE HAS BEEN WORKING IN THE FILM INDUSTRY HIS ENTIRE LIFE, AND HAS SUCH MOVIES AS *ROBOCOP, LONESOME DOVE, AND JIMMY NEUTRON: BOY GENIUS* TO HIS CREDIT, AS WELL AS TV SHOWS LIKE *WALKER, TEXAS RANGER.* DEUCE HAS BEEN USING COMPUTERS SINCE HE WAS 9, AND STARTED OFF WRITING HIS OWN GRAPHIC PROGRAMS. HE IS A UNIQUE COMBINATION OF PHYSICAL KNOWLEDGE AND VIRTUAL KNOW-HOW.

BY ERIC S. KELLER

DISAPPEARING/REAPPEARING ACT: ANIMATING TEXTURES IN MAYA

When I create an animation in Maya, I always keep an eye on making things easy to change down the road. To that end, I use the connection editor, expressions, and set driven keys as much as possible, and as few keyframes on the time line as I can get away with. When I'm animating textures in complex hypershade networks, as you'll see in this tutorial, this approach becomes a real sanity saver. If I were to just set keys on various attributes in a complex texture network, I'd be sure to end up hunting for these keys in a zillion nested panels when changes become necessary.

The goal for the animation in this tutorial was to have a character "beam in" or reveal itself in a scene as a plane moves down the Y axis. Sort of my own variation of the Star Trek transporter beam, but with a few twists. The effect is created almost entirely with animated textures on a character that is a single polygon mesh.

First, let's take a look at the character shown in figure 1. I have created a very simple polygon model of a little dog in a space suit. She is a single piece of polygonal geometry, including her eyes. This model is not ideal for extensive character animation, but she'll work just fine for demonstrating animated textures. She's based on my dog Daisy, who in real life also looks perpetually worried.

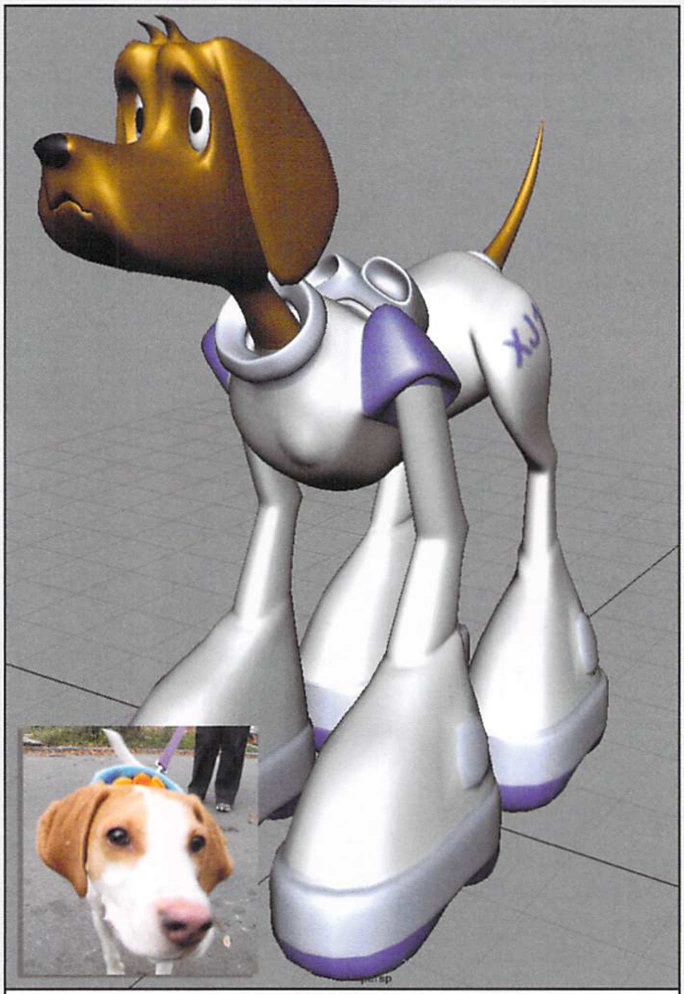


FIGURE 1 – A SCREENSHOT OF THE SPACE DOG, INSET IS A PHOTO OF HER INSPIRATION

I will setup the textures so that as a NURBS plane moves down in the Y-axis, she will begin to appear out of thin air; everything above the plane will be visible and everything below will be invisible. When I was developing the idea for this animation, it occurred to me that one way to achieve this effect would be to animate a ramp on the transparency channel of the dog's shader. However, I didn't want to animate the color channel of the shader, just the transparency. Furthermore, the color would have to be mapped to the UVs on the body in one way, and the transparency would be mapped in another. So, to avoid this problem, I set up multiple UV coordinate sets for the different texture channels.

SETTING UP MULTIPLE UV SETS

1 I assigned a Blinn texture to the model and then spent some time mapping the UVs in the UV texture editor. The result is a pretty standard UV texture map, which you can see in figure 2. It's not the most efficient map in the world but it works well enough for this demonstration. I won't go into detail on creating these UV texture coordinates in this tutorial since I believe this topic has been covered in plenty of others.

2 I assigned a Photoshop Texture to the Blinn shader for the color and the specular color channels. One of the new developments in Maya 6 is a node for creating a Photoshop file texture with layers for selected channels. I recommend

checking out Maya help to see how to do this. If you're using an older version of Maya, you can use a more old-fashioned approach with multiple image files, just as long as you end up with file textures for the color and specularity channels.

3 Once I had the textures painted and mapped on my dog model, I renamed this UV set by choosing (in the modeling menu set) *Edit Polygons>Texture>Rename current UV set*, and typing in the name "dog_color" in the dialogue box.

4 In the Blinn texture assigned to the model, I created a ramp for the transparency channel. In the texture option box, I made sure it was set to "Normal" as opposed to "As Projection" or "As Stencil." I wanted the ramp to be black and white with no interpolation. Where the ramp is black the texture will be opaque; where it is white, the texture will be transparent.

5 I deleted the top color from the ramp and set the other two to black and white (the values should be 0 and 1, respectively). I arranged the colors in the ramp so that the black position is at .5 and the white position is at 0 as shown in figure 3.

6 To see how this ramp is applied, I opened the attribute editor for the Blinn texture and chose the transparency channel for the Hardware Texturing preview. The default resolution should be OK, but sometimes it requires setting to medium or high to see it correctly in the perspective view.

7 When I did this, it became immediately apparent that the ramp was mapped all wrong. Take a look at figure 4 to see how badly this came out. The new transparency ramp used the same UV coordinates as the color channel, but what I actually wanted is for the ramp to cut across the model in an even horizontal fashion halfway down the model. To do this, I created a new set of UV coordinates. I selected the model and chose *Edit Polygons>Texture>Cylindrical mapping*, and in the options I selected the check box

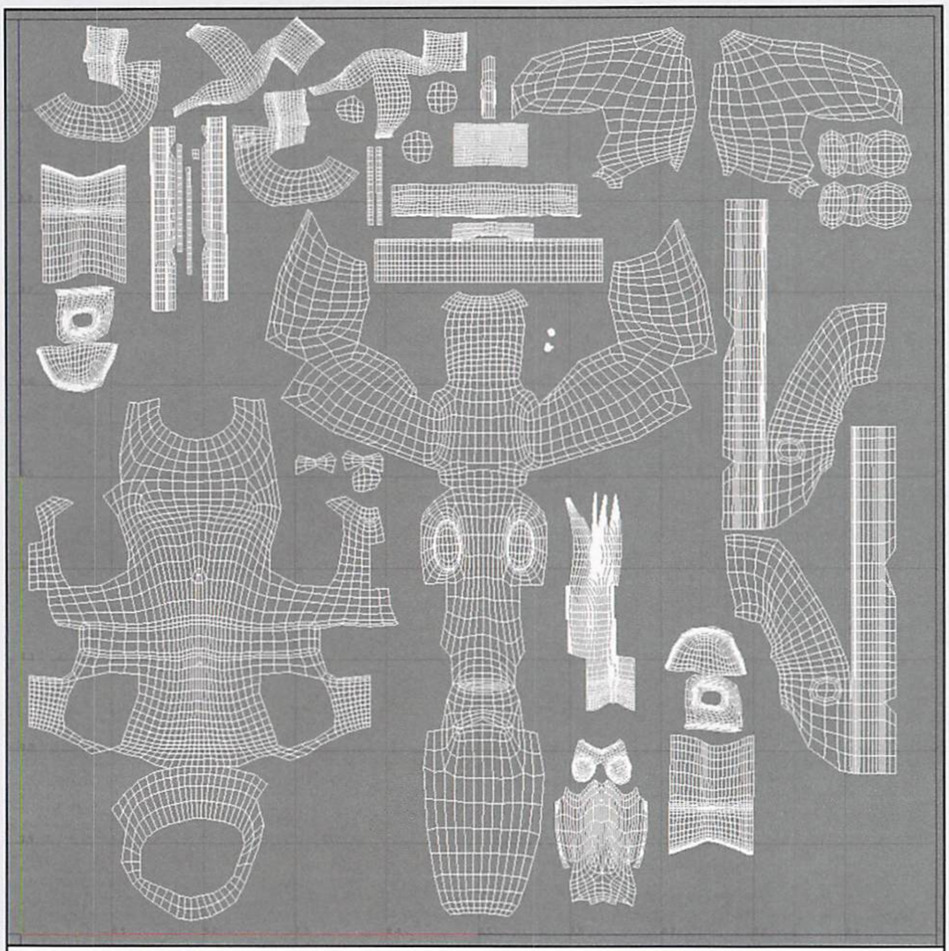
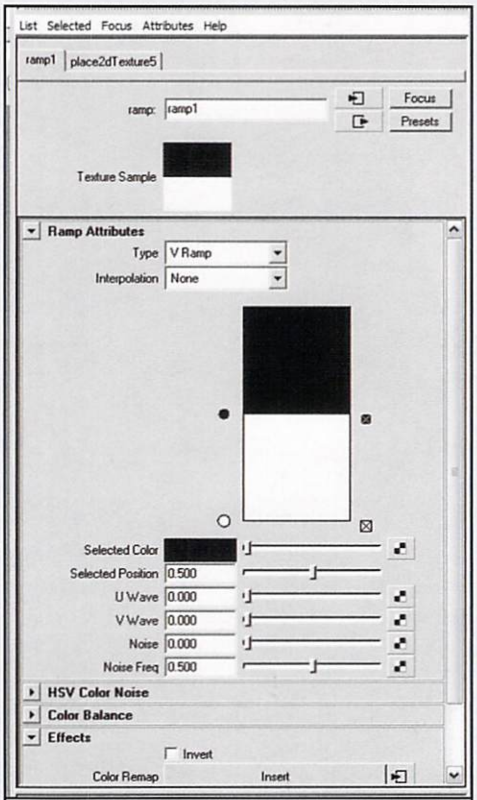


FIGURE 2 – THE UVS FOR THE MODEL AS SEEN IN MAYA'S UV TEXTURE EDITOR



ramp: ramp1

Ramp Attributes

Type: V Ramp

Interpolation: None

Selected Position: 0.500

U Wave: 0.000

V Wave: 0.000

Noise: 0.000

Noise Freq: 0.500

Effects

☐ Invert

Color Remap

FIGURE 3 – A SCREENSHOT OF THE RAMP THAT IS CONNECTED TO THE TRANSPARENCY CHANNEL

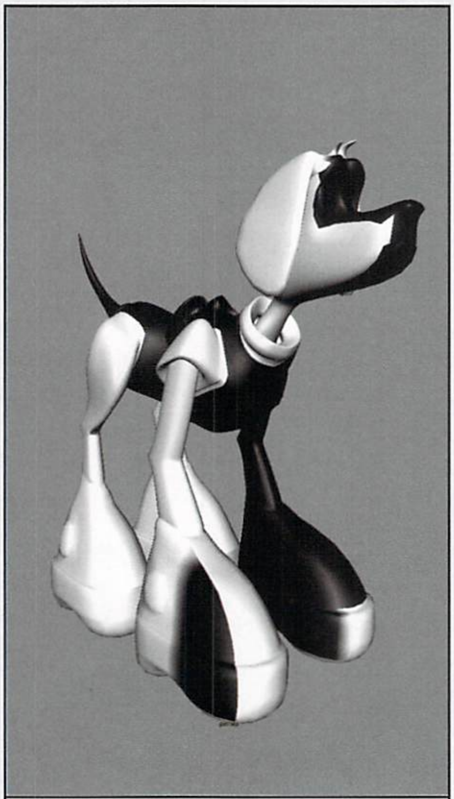


FIGURE 4 – RAMP IS CONNECTED TO THE SAME UV SET THAT CONTROLS COLOR TEXTURE PLACEMENT

for "Create New UV Set." I named these new coordinates "dog_transparency." Then, I assigned the ramp texture to this new set of UV coordinates.

8 Under *Window>Relationship Editors> UV Linking*, I chose UV-Centric, which opens up a new dialogue box. On the left side, it shows the UV sets, and on the right side it shows the channels for the Blinn textures and their assigned textures. When the "dog_color" UV set is selected, all the textures on the right side are highlighted. I selected the "dog_transparency" UV set on the left, and the ramp1 texture on the right. This automatically disconnects the ramp from the "dog_color" set and assigns it to "dog_transparency." Now the mapping of the ramp on the model should look more like what I want. I then opened up the UV texture editor to do some clean-up on this UV set.

9 The UV mapping for the transparency channel doesn't have to be terribly complicated. I selected the UVs and scaled them down so that they fit in the upper right hand corner of the grid. Any stray UVs caused by the mapping I selected and cleaned up by moving them in. You can see how the UV set looks in figure 5. In the figure, I have the UVs fitting roughly between 1 and 0 in U, and .25 and .75 in V. Once this is done, the upper half of the dog should be black and the lower half white.

Rendering out a still using Maya's software rendering should show the bottom half of the dog as invisible. However, I noticed that, since there is a file texture mapped to specularity, and this channel is connected to the "dog_color" UV set, the bottom half of the dog is still reflecting light, like transparent plastic. See figure 6. I'll get to that in a moment. First, I'll describe how I set up the controls for the animation of the transparency channel.

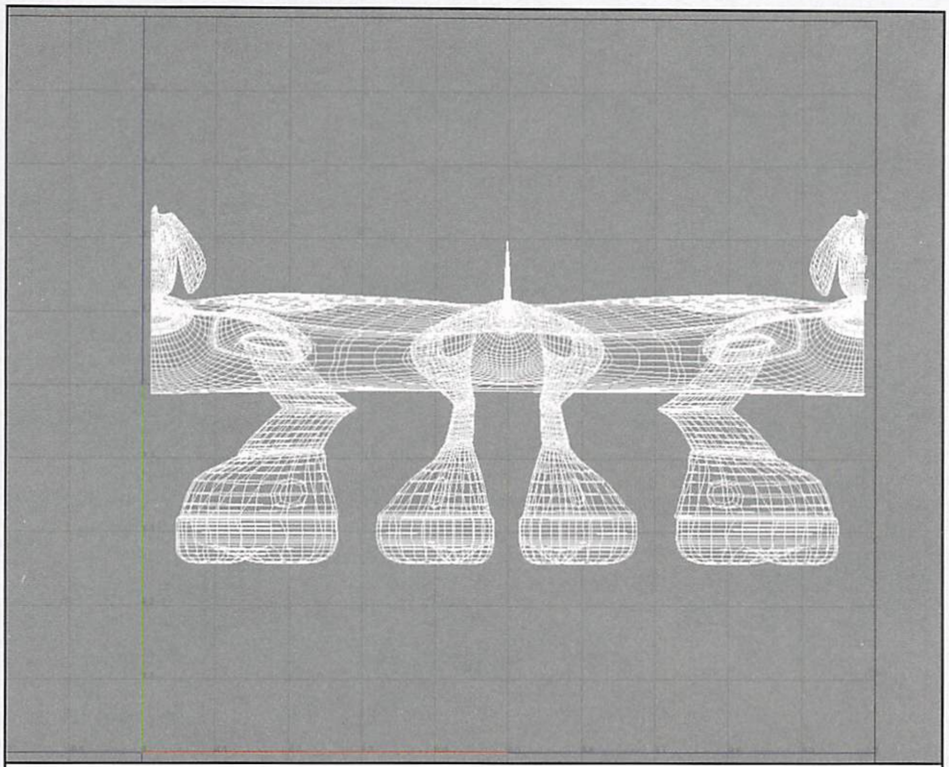


FIGURE 5 – A NEW UV SET HAS BEEN CREATED TO CONTROL THE TRANSPARENCY MAP VIA A CYLINDRICAL TEXTURE PLACEMENT

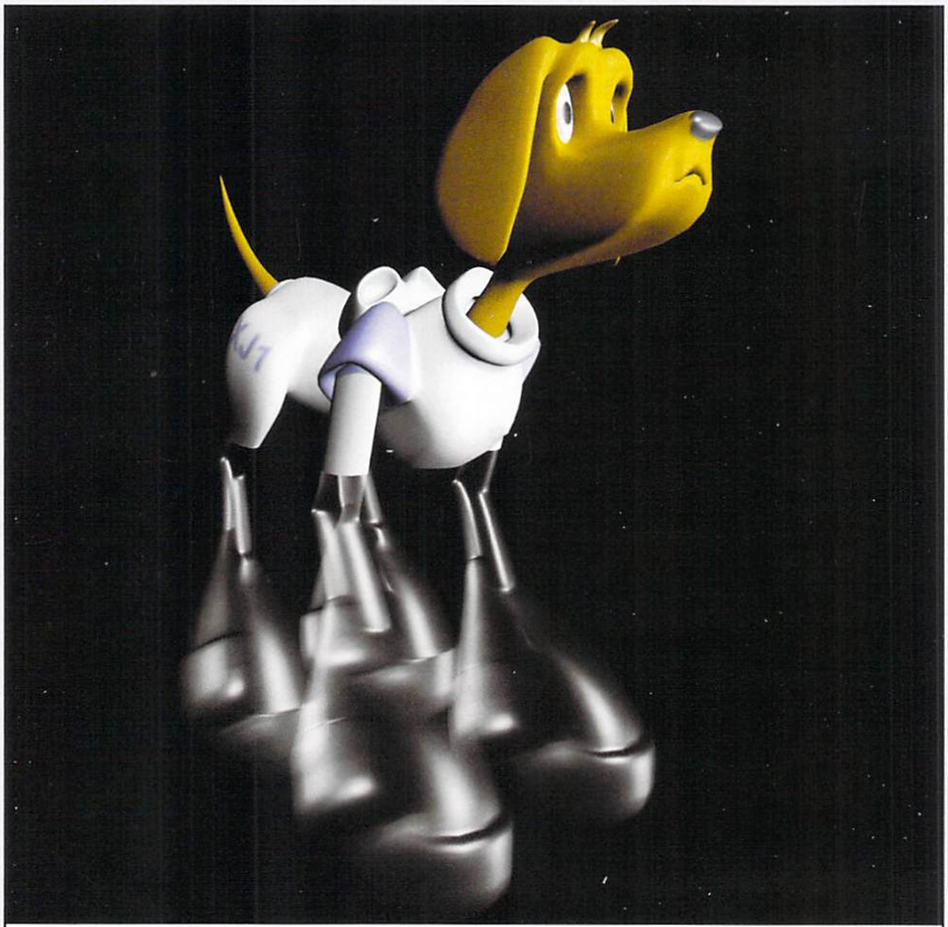


FIGURE 6 – THE TRANSPARENCY RAMP IS NOW MAPPED CORRECTLY BUT THE FILE TEXTURE IN THE SPECULAR CHANNEL IS CAUSING SOME WEIRDNESS

ANIMATING THE TRANSPARENCY TEXTURE

10 I created a NURBS plane and scaled it so that, from the top, the border encompassed the dog model. What I wanted to do was to create a relationship between the plane's Y translation and the position of the black color border on my transparency ramp. I could have set some keyframes on to the position of the black border on the ramp in the attribute editor, but, as I stated before, I wanted to control everything by the Y translation of the plane. This way, if I decide to change the timing of the animation later on, I would only have to change keyframes on the plane's position. To do this, I used the Set Driven Key to establish the relationship between the plane's translation and the ramp's black color position.

11 I opened up the dialogue box for "Set Driven Key," which is found under the "Animate" menu in the animation menu set. I selected the NURBS plane and chose "Load Driver." It would then seem logical to load the ramp texture on the driven side of the Set Driven key box, and then select its color input position for the black border from the list of driven attributes so as to connect it to the translate Y of the plane on the driver side of the box, but Maya won't let you do this. The individual color input positions on the ramp don't show up in the attribute list when you load the ramp in the "Driven" half of the box (go figure!). The work-around I used was to load the transparency ramp's place2DTexture node in the driven part of the box instead of the ramp texture itself and use the plane's Y translate to drive the offset attribute on this texture placement node.

12 So to do this, I selected the NURBS plane in the perspective window, opened up the Set Driven Key, and loaded it as the driver on the left side of the box. Then I opened up the hypershade window and selected the place2DTexture node that controls the transparency ramp texture. With this selected, I pressed the "Load Driven" button on the right side of the Set Driven Key dialog box.

13 Then, in the side view window, I moved the plane up until it was just above the top of my model. Next, I went to the attribute editor for the place2dtexture node, I typed in .75 for the offset in V. Doing this will reposition the ramp so that the entire dog is white, which means it will render as transparent. In the Set Driven Key dialog box, I selected the translateY channel for the NURBS plane in the driver side of the box and the offset attribute for the ramp's place2Dtexture node in the driven side of the box, and then I hit the key button on the bottom of the Set Driven Key box. phew!

14 Next, I moved the plane down until it was just below the feet of the dog. In the attribute editor for the ramp's place2dTextureNode, I typed .25 in the offset box, which repositioned the ramp so that the entire dog is black (meaning that the dog will render as opaque/visible). Again I selected the translateY channel for the NURBS plane from the attribute list on the driver half of the Set Driven Key dialog box, and the offset attribute for the ramp's place2dTextureNode node in the attribute list on the driven half, and I hit the key button on the bottom of the box. Now a relationship between the plane's Y position and the ramp that controls the dog's transparency has been established. Moving the plane up and down makes this apparent in the perspective window - provided you have transparency selected as the hardware texturing mode in the Blinn texture's attribute box. A few test renders will also confirm the relationship between the plane's Y translation and the dog's transparency.

15 Now, the transparency and the color channels were rendering the way I wanted it to, but the specular reflection still showed up on the surface. This makes it look as though the dog is some kind of clear plastic that is being filled in with color as the plane moves down. It's a cool effect, but not what I want. To address this specular reflection problem, I thought about connecting the same ramp texture that controls the transparency to the Blinn's specularity channel. However, the specularity channel already has a file texture node connected to it, and, furthermore, the transparency's ramp texture is the reverse of what I want!

If I were to go ahead and hook up the same ramp that controls the transparency to the specular color channel of the Blinn shader, the white part of the ramp would make the dog both transparent and give her a white specular color, and the black areas of the ramp would make her opaque but with no specular color. What I want is the opposite of this situation: specular reflections on the visible/opaque parts of the model and no specular reflection on the invisible/transparent part of the model. My solution was to first create a reverse node in the hypershade (under General Utilities), and then connect the ramp's outColor to the reverse node's input connection. I did this in the hypershade by middle-mouse-button dragging the ramp node on top of the newly created reverse node and selecting "other" from the pop-up menu. This opened the connection editor, where I connected the ramp's outColor to the reverse node's input. So now I have essentially created a reversed ramp that is controlled by the Y translation of the NURBS plane without actually having to create a new ramp texture or any more set driven keys and it will render the way I want it to.

16 To complete the connection to the specular color channel without losing the specular channel's connection to the file texture, I connected the reverse node's output to the colorGain on the specular color channel's file texture node. This essentially layers the reversed ramp texture on top of the specularity channel's file texture. Now where the ramp is black, the specular file texture is black, but where the ramp is white, the

file texture shows up just fine. Figure 7 shows this part of the shading network. Phew again!

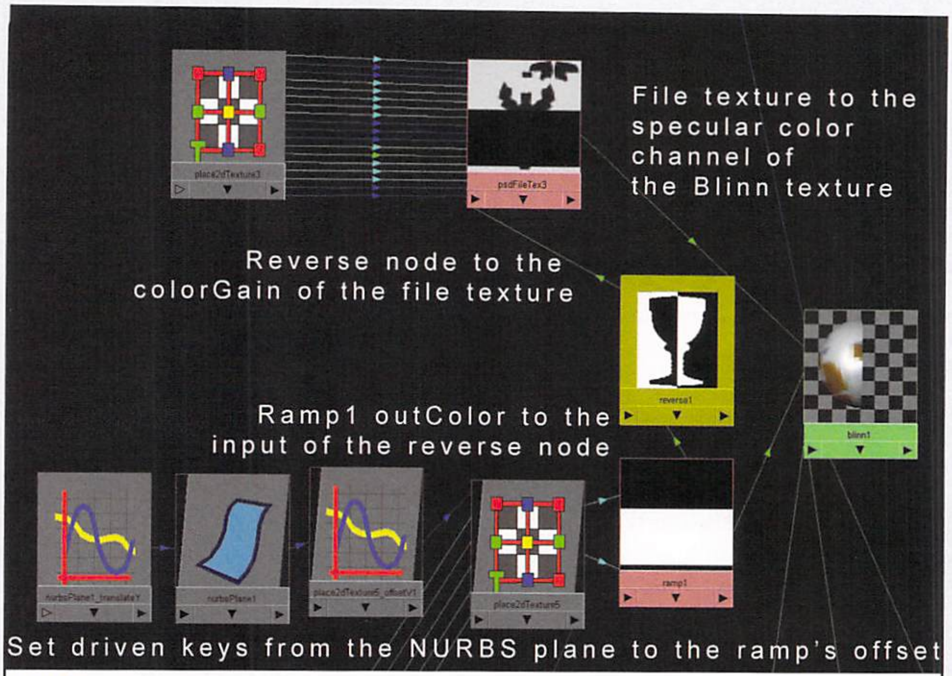
INCANDESCENCE

As an additional touch, I wanted to create a bright glowing beam at the intersection of the plane and the dog model. The setup will be very similar to the transparency ramp, but since I wanted a thin beam, I ended up creating a new ramp texture for incandescence.

17 To set this up, I created another ramp in the incandescence channel of the Dog's Blinn texture. This new ramp has a thin beam of green in the middle at the .5 position.

18 Just as in the case of the transparency, I want this incandescent beam to be mapped as a cylindrical projection. The easiest way to do this was to hook it up to the same UV set as the transparency. I opened up the UV Linking relationship editor in UV centric mode. I selected the dog_transparency set on the left and the ramp under the incandescence on the right. Now, when I render the scene, there should be a bright glowing green band running midway around the dog but, as of yet, the movement of the plane does not affect the placement of this band.

19 To fix this, I opened up the connection editor and the hypershade. I selected the place2Dtexture node that controls the placement of the transparency map and loaded it on the left side of the connection editor. I then selected the place2Dtexture node that controls the incandescence ramp and loaded it on the right side of the connection editor. On the left side of the connection editor I selected OffsetV from the attribute list, and then connected to the OffsetV on the right side. Now the Y translate of the plane controls the offset of the incandescence ramp indirectly via the set driven keys I already placed on the transparency ramp.



File texture to the specular color channel of the Blinn texture

Reverse node to the colorGain of the file texture

Ramp1 outColor to the input of the reverse node

Set driven keys from the NURBS plane to the ramp's offset

FIGURE 5 – A VIEW OF THE HYPERSHADE SHOWING THE RAMP'S CONNECTION TO BOTH THE TRANSPARENCY AND THE COLORGAIN ON THE SPECULAR CHANNEL'S FILE TEXTURE. NOTE THE REVERSE NODE!

MENTAL RAY

If you're using Mental Ray to render the image, the render may abort and give you an error such as *““maya_place2dtexture” has no member “offset.”* A quick workaround that Mental Ray prefers, involves using an expression on the position of the black color of the ramp. (If you're using Maya's software rendering engine, skip ahead to step 23.)

20 I removed all of the set driven keys on the ramp textures controlling transparency, incandescence, and color gain on specular color texture. I also went into the UV texture editor and normalized the dog_transparency set so that the UVs fit perfectly between 0 and 1 in both U and V. The easiest way to do this is to select all the UVs and, in the UV texture editing menus, choose Polygons> Normalize UVs. Make sure that “collectively” is checked and “preserve aspect ratio” is unchecked in the options.

21 Then, I moved the plane so that it was right at the very top of my model. In this case, that would make the Y translation value = 6.885. I loaded my transparency ramp into the attribute editor, right-clicked on the position for the black color on the ramp, and chose “create new expression” from the pop-up menu.

22 The expression I entered was ramp1.colorEntryList[2].position=nurbsPlane1.translateY*.14524. Why .14524? Well, I want the position of the black color in the ramp to be at 0 when the plane is at 0 and when the plane moves to the top of the model (6.885), I want the black color to be at 1. 1/6.885 = .14524…(I rounded). So now the black color on the ramp should move with the plane just like in the earlier setup, and this won't cause Mental Ray to give up during the render. A similar expression can be put on the color positions of the incandescence ramp.

TEXTURING THE PLANE

The rest of this tutorial picks up from step 19 and uses Maya's software rendering engine instead of Mental Ray, most of these following techniques should work for both renderers.

So now I've solved my main problem. Moving the plane up and down creates the effect of the dog appearing or disappearing. However, the plane

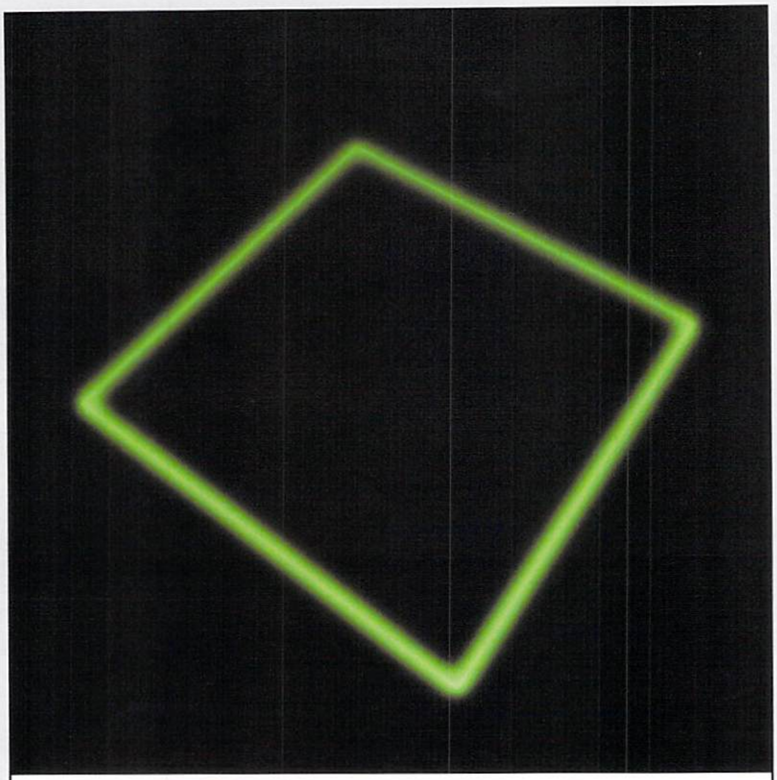


FIGURE 8 – THE PRIMARY GRID TEXTURE ON THE OUT GLOW COLOR OF THE PLANE'S SHADER

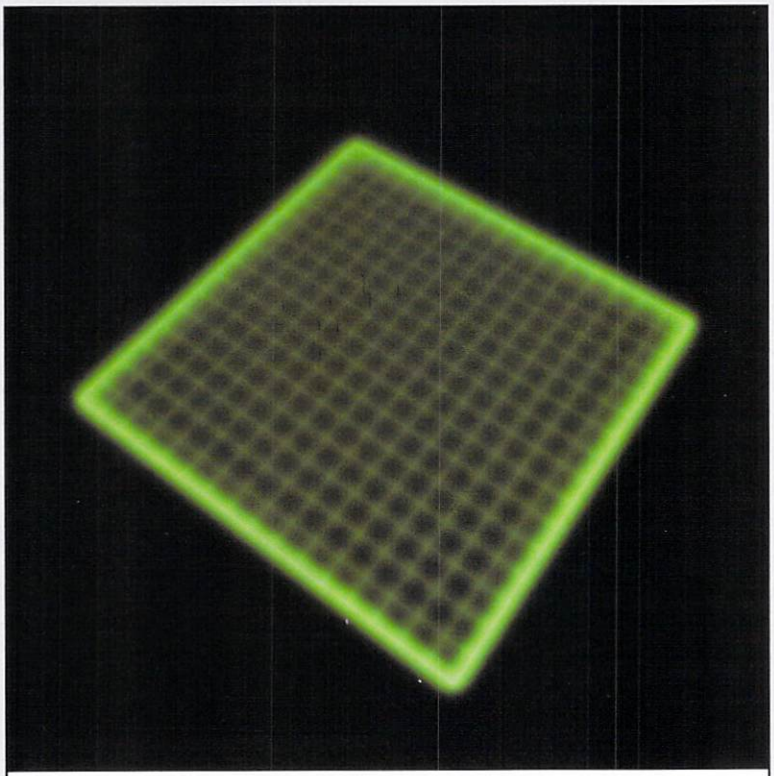


FIGURE 9 – THE SECONDARY GRID TEXTURE APPLIED ON TOP OF THE PRIMARY GRID TEXTURE

looks kind of dull, so I decided to add an animated texture just to make it look more interesting.

23 I assigned a surface shader to the plane and made the color black and completely transparent (i.e. white in the transparency channel).

24 I assigned a grid texture in the out glow channel.

25 I made the line color green and the filler color black because I wanted to make a border that just illuminated the edges of the plane. Since my plane was perfectly square, a grid texture worked just fine.

26 I set the U and V width of the grid to .005. In the place2DtextureNode for the grid, I set the repeat UV attributes to 1. Rendering this shows a nice glowing green border that goes around the plane. See figure 8.

27 I went back to the grid node and, in the filler color, I created another grid node. This second grid has a duller green for the line color and black for the filler color.

28 In the second grid's place2dtexture node, I set the repeat UV attributes to 16. Now the plane renders as a growing green grid with a bright border. See figure 9.

The last detail I wanted to add to this glowing grid plane was a pulsing glow that moves across the plane as it moves.

29 I went back to the first grid and, in its color offset channel under the color balance attributes, I created a V Ramp.

30 I made the ramp black on the top and bottom, and created a dim gray band in the middle.

31 In the color channel for this band, I created a fractal texture.

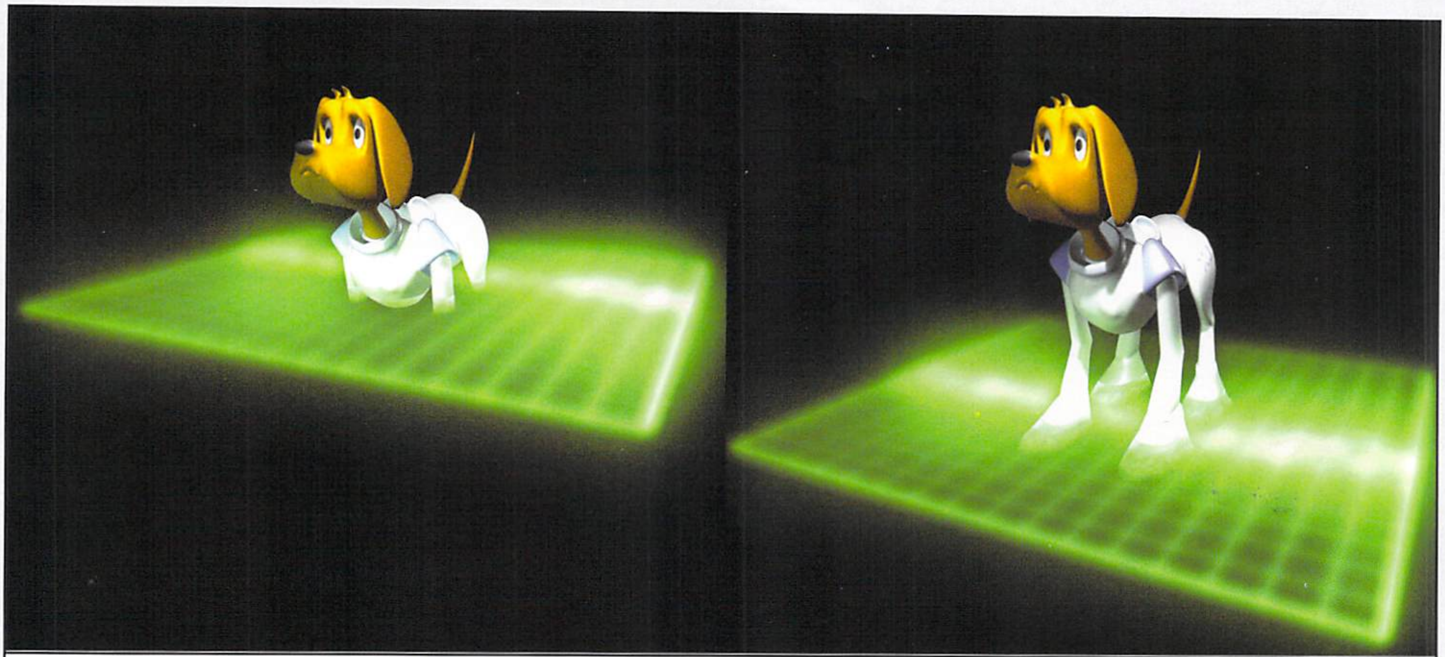


FIGURE 10 – THE FINAL ANIMATION SHOWING THE SPACE DOG BEAMING IN

32 To make the fractal green, I went into its attributes and chose a dim green color for the color gain under the color balance attributes.

When I rendered this scene, I had a green glowing grid with a bright fractal beam in the middle. You can see this is the final rendered stills of the animation in figure 10.

33 To animate the fractal beam moving across the grid, I used a set driven key again to set up a relationship between the NURBS plane's Y translation and the V offset of the place2DTexture node that controls this dim green fractal ramp. This time, I set the V offset to 0 when the plane is at the very top of the dog and 1 when the plane is down to about the bottom of the ears.

34 I selected this place2Dtexture node, opened the graph editor, selected the offsetV curve and chose "cycle" for the pre and post infinity options under the curves menu. Now the green band will constantly zip across as the plane moves down, and once again, this motion is directly related to the Y position of the plane, so I can keep the keys on the timeline to a minimum. If I change the rate at which the NURBS plane moves down, this glowing beam will speed up or slow down accordingly. Mental Ray does not like the connection to the offsetV on the place2dTexture node, so you may want to substitute this with an expression like the one described earlier.

Setting up this effect takes some work, but the animation is easy. All you have to do now is set key frames on the NURBS plane's Y translation. If I decided that I want the dog to disappear instead of appear, I can just animate the plane moving up instead of down.

I hope you got a few ideas from this tutorial on how to economize future animations using set driven keys, multiple UV sets, the connection editor, and some simple expressions. Although the techniques in this example are fairly specific to this animation, you should be able to figure out how to use similar techniques to create different effects. Some ideas that come to mind include a glass filling with a colored liquid, or some kind of vaporization ray. A QuickTime movie of the final animation is available in the resource section of the HDRI3D Magazine website at www.hdri3d.com/resources.html.

ERIC S. KELLER HAS SPENT THE LAST SIX YEARS AS A SCIENTIFIC ANIMATOR AT THE HOWARD HUGHES MEDICAL INSTITUTE IN CHEVY CHASE MARYLAND. WHEN HE'S NOT RENDERING MOLECULAR DATA AT WORK, HE ENJOYS CREATING ANIMATIONS AT HOME IN WASHINGTON DC. HIS TRAINING IS IN MUSIC AND ART. HIS PRIMARY TOOLS ARE MAYA, ZBRUSH, PHOTOSHOP, AND AFTER EFFECTS. INSPIRATION COMES FROM HIS WIFE ZOE, THEIR TWO DOGS, AND WAY TOO MUCH TV.

TSUNAMI

EDIT. ANIMATE. FAST.

TSUNAMI Workstation for LightWave 3D®

We have taken the most complete and flexible software solution for 3D graphics and animation and coupled it with our dual Xeon® TSUNAMI Force to create the best solution available for the professional animator. Proven for years in television, film, and games, LightWave 3D [8] is versatile enough to make the transition to and from all kinds of projects. It is a full, robust program that includes many of the tools that other packages require to be purchased separately: soft-body dynamics, particles, hair and fur, unlimited render nodes, and more. The ultra-powerful TSUNAMI can be custom-built to your ideal specifications from a long list of components. Price yours today at www.sharbor.com/build-yours/

Cloth and Vehicle Dynamics
This tutorial explores LightWave [8]'s ClothFX tool set and shows you how to spice up your cloth and vehicle animations. You'll learn how to animate believable flags, footprints, clothing, hovering vehicles, motorcycles, cars, and more! 36.99

Bob's LWConnect

LWConnect
This simple to use plug-in joins VTEdit with LightWave 3D. Harness the power of LightWave to shake, twist and bend multiple video tracks without having to know how to animate! Alpha-channel DVE's, picture-in-picture, unlimited layers, and more 279.00

X-Dof 2.0
X-Dof is an expert depth-of-field solution, production proven and fully optimized for highend 3D content creation. This plug-in for LightWave features envelopes on every parameter, soft depth-of-field effects, custom pixel and surface intensity, advanced motion blur, and more! . .199.00

NEW!
Instant VT[4]
Learn VT[4] in just over 2 hours

Instant VT[4]
Be ready to tackle your next edit or production project with confidence. Whether you're new to VT or a seasoned veteran, this training is a must-have! Class on Demand will show you how to use VT[4] to meet your production and post-production needs 69.99

Industrial Modeling: Product Creation Volume 1
Learn to model real-world objects with Larry Shultz, an industry-leading VFX artist. This 3+ hour video is designed to give you a real world example of a 3D for print or 3D animation job, and how to go about working through the project from start to finish .27.99

Mirage for LightWave
Mirage brilliantly combines real-time paint, animation, and effects functions into a single, easy-to-use product. LightWave 3D users especially benefit from Mirage's ability to paint on Lightwave UV maps directly in the Mirage Project Window, and more! 449.00

NEW!
WAVE GUIDE
WAVE GUIDE
WAVE GUIDE
WAVE GUIDE

WaveGuide
Four valuable volumes take you through LightWave's toughest challenges! Go from learning the basics to 2D animation, dynamics, compositing, and more. Taught by renowned LightWave instructor and animator Patrik Beck 149.99
Individual WaveGuides .39.99

ButterflyNetRender 3.0
Manage your network rendering with ease and sophistication! Now supports LightWave 3D [8], 3DSMax 6+, and Maya 6. New features include: HTML Viewer/Internet support, 'FrameCycler' support, post commands, scene linking, AVI building, Maya 6 MR support and more 579.99

Terms: POs accepted from schools and government agencies • All checks require 5-7 days to clear • Defective products replaced promptly • RMA number required for all returns • Returns accepted within 15 days, in original packaging, postage prepaid, undamaged • Opened software is not returnable • Shipping charges are not refundable • Returns are subject to a 15% restocking fee • Not responsible for typos • Prices are subject to change

NewTek™ Authorized Reseller

Authorized Reseller

Information
262-548-8120

Tech Support/RMAs
262-548-8157
Mon-Fri 8:30am-5:00pm CST

Fax: 262-548-8130

Safe Harbor Computers
W226 N900 Eastmound Dr.
Waukesha, WI 53186

www.sharbor.com

© 2005 Safe Harbor Computers. All rights reserved. TSUNAMI and the TSUNAMI logo are trademarks of Safe Harbor Computers.

YOUR SOURCE FOR LIGHTWAVE ANIMATION PRODUCTS!

www.sharbor.com

800-544-6599

SAFE HARBOR COMPUTERS

BY ALEXANDER SHAREEF

VISUAL-EFFECTS TRICKS!

PORTAL PLASMA

There is no one correct way of doing something. This is one of the most important things a Visual Effects artist can learn. As a Visual Effects artist, you must learn this early on, because in order to create amazing magical spectacles, the simplest of ways are often the best. It may not be the greatest way of doing it, but generally it's the fastest…and most importantly…it works.

It's the simple little tricks that are the foundation and building blocks for all special effects work. Small little things that are broken down then reassembled together to create something far greater than the sum of its parts. A Visual Effects artist is really just a problem solver, figuring out ways of fooling the eye into believing it's something completely different…something wonderful, instead of something simple.

Take, for example, the scrolling texture on a cylinder trick…at first, you might think something as simple as that wouldn't be convincing as anything…but for a long while, it was the main way of doing an engine's thrusters for a plane or spaceship.

If you take this cylinder trick one step further and pull down the sides of the cylinders, it will give the appearance that the texture is now moving outwards from the center in a circle. This is an incredibly simple but useful trick, and one that I use all the time at work; it is very good at making things look like ethereal energy.

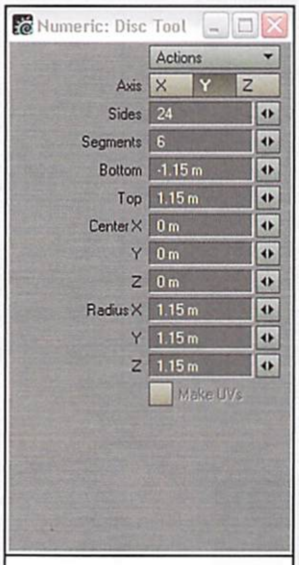
What I'm going to do in this tutorial is show you this particular trick, and in doing so, make some kind of magical portal. Hopefully, this should be a good introductory tutorial to the world of effects. Though it's actually far easier to do this trick in other programs like Maya or 3ds max, I'm going to take you through it in LightWave, since that's the package I started learning with some six years ago and I'm partial to its coolness.

TUTORIAL STEPS

First of all, let's make a cylinder like in *(Fig 1)*. I've kept it to 24 sides so as to keep the roundness of the circle, and six segments gives us plenty to play with for UVs later *(Fig 2)*.

Delete the top and bottom faces of the cylinder as shown in *(Fig 3)*. We won't need those.

Now before proceeding any further, let's make sure to apply some UV-mapping to the cylinder. Create a new Texture Map *(Fig 4)*, and call it something meaningful. I named it "CylinderUVs." Don't forget to change the mapping type to Cylindrical *(Fig 5)*. For those that don't know what UVs are, what



Axis: X Y Z
Sides: 24
Segments: 6
Bottom: -1.15 m
Top: 1.15 m
Center X: 0 m
Y: 0 m
Z: 0 m
Radius X: 1.15 m
Y: 1.15 m
Z: 1.15 m
☐ Make UVs

FIG 1 — CREATE A CYLINDER

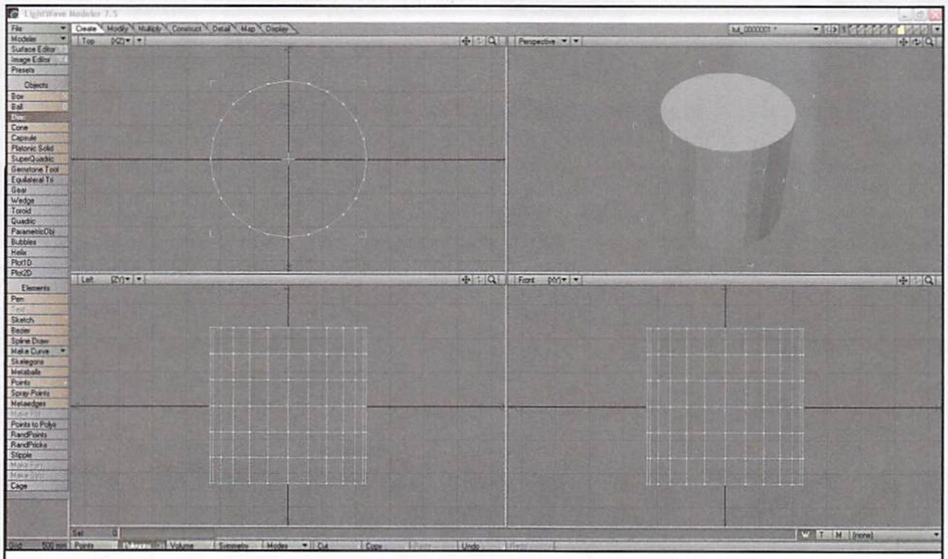


FIG 02 — YOU'D NEVER BELIEVE THIS CYLINDER WAS THE PORTAL

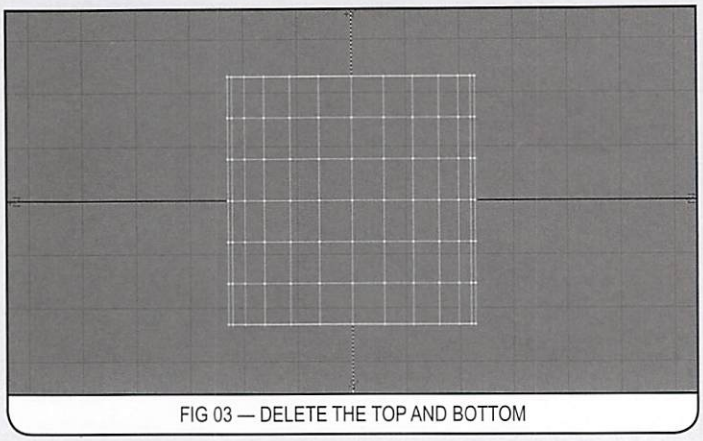


FIG 03 — DELETE THE TOP AND BOTTOM

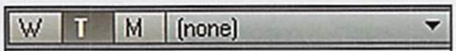
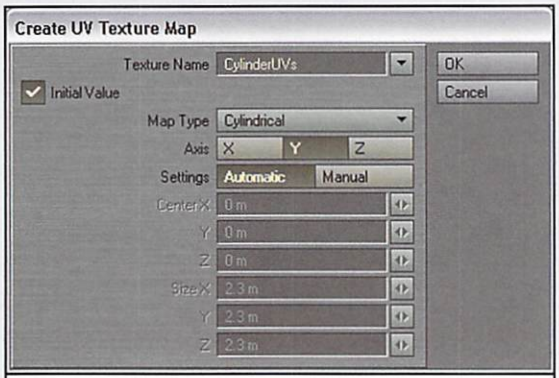


FIG 04 — CREATE A UV TEXTURE MAP



Create UV Texture Map

Texture Name: CylinderUVs

☑ Initial Value

Map Type: Cylindrical

Axis: X Y Z

Settings: Automatic Manual

Center X: 0 m

Y: 0 m

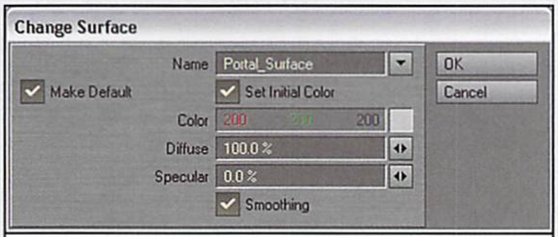
Z: 0 m

Size X: 2.3 m

Y: 2.3 m

Z: 2.3 m

FIG 05 — GIVE THE UV MAP A NAME THAT MAKES SENSE



Change Surface

Name: Portal_Surface

☑ Make Default

☑ Set Initial Color

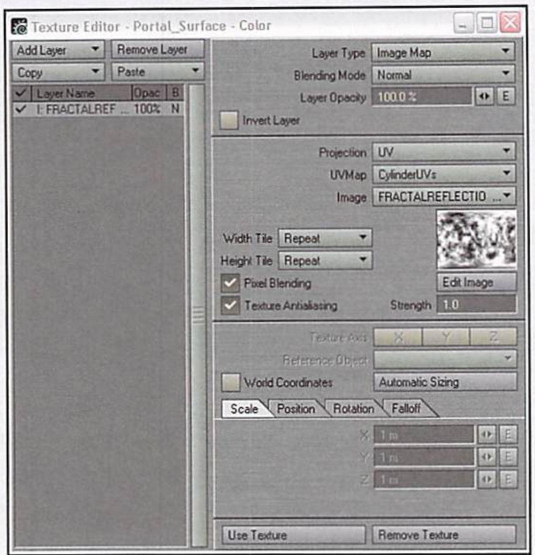
Color: 200 200 200

Diffuse: 100.0 %

Specular: 0.0 %

☑ Smoothing

FIG 06 — CREATE A NEW SURFACE



Texture Editor - Portal_Surface - Color

✓	Layer Name	Opac	B
✓	I: FRACTALREF ...	100%	N

Layer Type: Image Map

Blending Mode: Normal

Layer Opacity: 100.0 %

☐ Invert Layer

Projection: UV

UVMap: CylinderUVs

Image: FRACTALREFLECTIO ...

Width Tile: Repeat

Height Tile: Repeat

☑ Pixel Blending

☑ Texture Antialiasing

Strength: 1.0

Texture Axis: X Y Z

Reference Object:

☐ World Coordinates

Scale: X: 1 m, Y: 1 m, Z: 1 m

FIG 07 — FOR NOW, JUST USE "FRACTALREFLECTIONS"

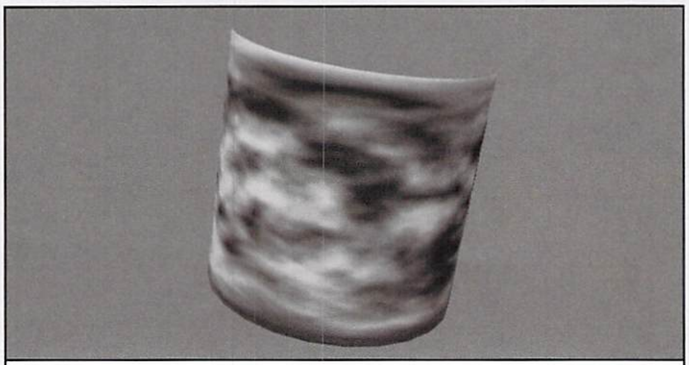


FIG 08 — WITH A TEXTURE ON, IT LOOKS MORE LIKE A PORTAL ALREADY

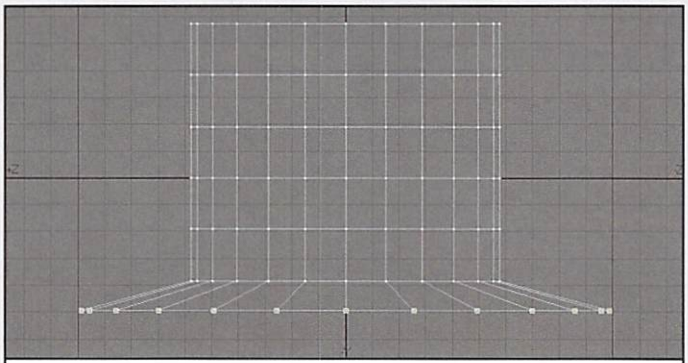


FIG 09 — SELECT EACH ROW IN TURN AND SCALE THEM OUT

we've done here is basically just wrap a flat texture around the cylinder. This is useful, because no matter what happens to the geometry, that texture will conform to each of the faces and can be unwrapped at any time. Imagine it as if you've just wrapped an invisible piece of paper around a coke can with superglue...even if you bend or squash the can; the paper will bend with it since it's glued to the sides. This is really the heart of the effect, and with UV scrolling can be very powerful. It's simple, but quite effective.

To better see this, how about we actually texture the cylinder. Hit q to bring up the surfacing panel, and call the surface something meaningful, like "Portal_Surface" *(Fig 6)*. Open up the Surface Editor, and proceed to add a texture to the Color channel. Choose "UV" as the projection type, and then under the UVMap tab, choose the one we created earlier, "CylinderUVs." For the moment, just choose the standard "FractalReflections.tga" that comes with LightWave as the texture *(Fig 7)*. We'll be creating our own custom texture later, but use this one for the moment so you can get an idea of what's happening with the UV texturing *(Fig 8)*.

So, now comes the interesting part...Let's do exactly what we imagined earlier and squash the cylinder down and out. Grab the bottommost points, and scale them out a bit *(Fig 9)*. Take the next row up and do the same, this time pulling them down a bit as well. Continue this until you have something similar to *(Fig 10)*. Modify it further, pulling in the topmost points (now the centermost) to close that gap in the middle as much as possible without stretching the UV mapping too much. Hopefully by the end of it all, you'll be left with something like in *(Fig 11)*.

Did you notice how the texture stayed 'super-glued' to each of the faces, and conformed to the new shape? It now appears squashed down, and looks somewhat portal-like! The cool thing is we can unwrap the UVs at any time to go back to their original flat shape. Or better yet, how about scrolling/animating the UVs so that they appear to flow from the center outwards?

There are a couple more things that need to be done first before we can move on to the UV scrolling. First of all, I'm going to paint the vertices with some colors. This is completely voluntary, and not necessary for the final portal, but as you'll see along the way it is quite a useful and easy tool for adding gradient-like colors quickly to the portal.

Under the "Map>Weight&Color" tab, select "Vertex Paint." You'll be greeted with what at first seems a very complicated window, but it's actually quite straightforward *(Fig 12)*. Under the edit menu, select "Create Vertex Color Map," and give it a name like "Portal_VertexMap" *(Fig 13)*. Now everything is pretty much already set up the way we want it to be, but I'll run through the important parts of the tool.

What the Vertex Paint tool does is exactly what it sounds like – it paints colors onto the individual points of the geometry. So, by doing this, each point will have their own color information that can be used in addition to the texture map.

Under the Color tab, you'll see the Brush Size set to 20, which is OK for the moment and can be changed on the fly by dragging the right-mouse button. You can select "Front," which tells it just to paint on the faces that the brush touches. Leave the brush shape as a circle. Further down, the Color Map that should be selected is the one we just created, namely "Portal_VertexMap." Keep it as RGBA, which will allow us to paint in the Alpha channel as well as Color channels. The "Color/Point" paint mode is perfect for what we're going to do, though please experiment with the other modes. The "REP" next to paint mode indicates anything we paint will replace whatever vertex information was already there. This can be set to Add, Subtract, or Erase, though keep it on Replace for this tutorial. The strength of the brush is akin to how much pressure you might normally paint with, so 100% is fine for us.

Finally, there are the actual colors that you can choose to paint with, either in RGB or HSV whichever you prefer. Our portal is going to be a blue base, but feel free to add in some purples and even greens to break it up a bit. I'll leave this up to your artistic eye, but I just painted each row a slightly different hue of blue. When you're done painting, you should have something like *(Fig 14)*.

This next bit is important…before you close the Vertex Paint, don't forget to click the "Save" button next to the Object name. The changes you've made here to the vertex colors will not be applied to the object in Modeler unless you do this. Also, you won't be able to see any changes in Modeler until you go into the Surface Editor and under the Advanced tab, select the "Portal_VertexMap" as the Vertex Color Map.

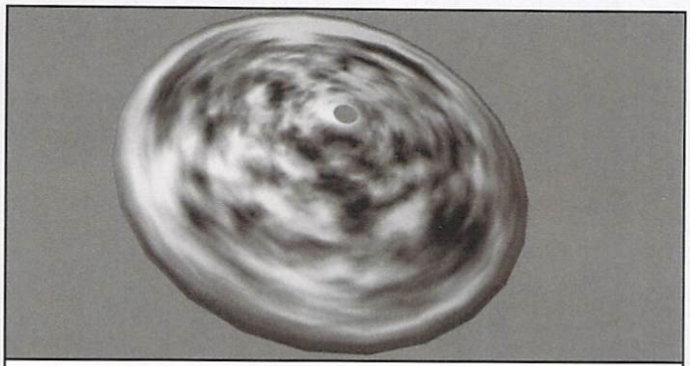


FIG 10 — THE CYLINDER IS NOW FLATTENED OUT

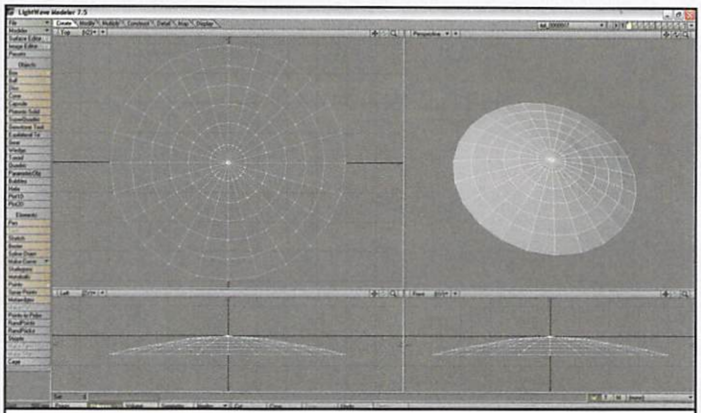
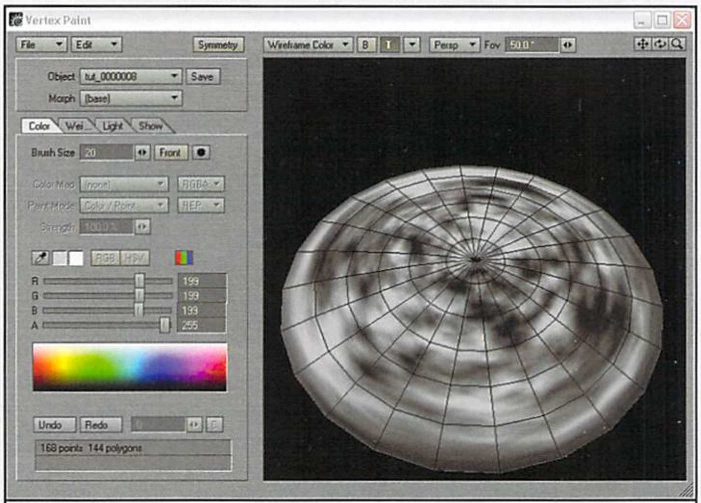


FIG 11— MAKE SURE THE TOP IS AT 0,0,0 IN MODELER



Vertex Paint

Object: tut_0000008

Morph: (base)

Brush Size: 20

Color Map: (none)

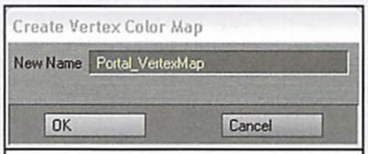
Paint Mode: Color / Point

Strength: 100.0 %

R: 199
G: 199
B: 199
A: 255

168 points 144 polygons

FIG 12 —THE VERTEX PAINT OPTIONS



Create Vertex Color Map

New Name: Portal_VertexMap

FIG 13 — MAKE A VERTEX COLOR MAP TO USE

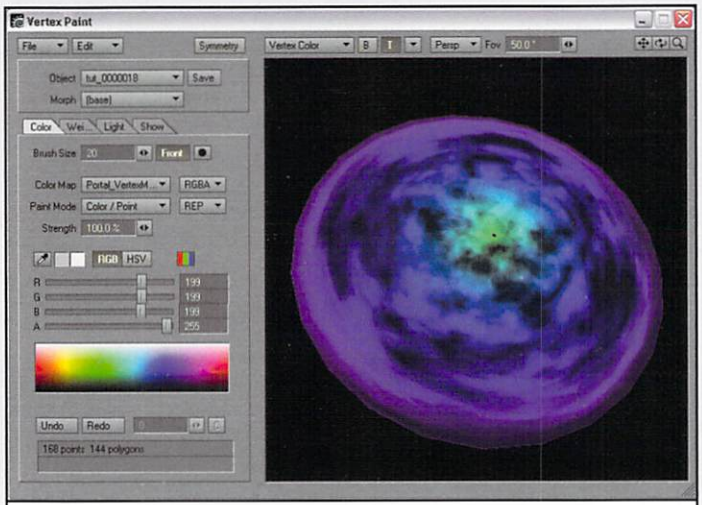
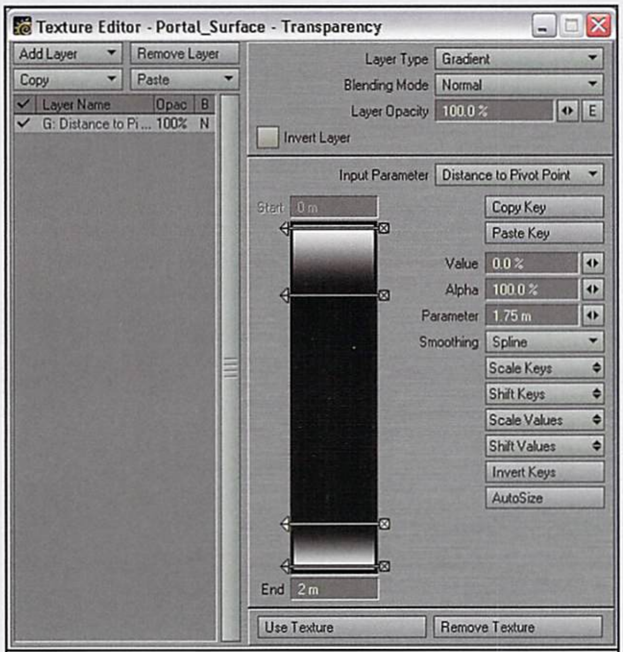

Vertex Paint
Object: tut_0000018
Morph: (base)
Brush Size: 20
Color Map: Portal_VertexM...
Paint Mode: Color / Point
Strength: 100.0 %
R 199
G 199
B 199
A 255
Fov: 50.0°
168 points 144 polygons

FIG 14 — BE ARTISTIC! PAINT IT IN ANY WAY YOU FEEL


Texture Editor - Portal_Surface - Transparency
Layer Type: Gradient
Blending Mode: Normal
Layer Opacity: 100.0 %
Input Parameter: Distance to Pivot Point
Start: 0 m
Value: 0.0 %
Alpha: 100.0 %
Parameter: 1.75 m
Smoothing: Spline
End: 2 m

✓	Layer Name	Opac	B
✓	G: Distance to Pi...	100%	N

FIG 15 — USE A GRADIENT TO GIVE THE EDGES A TRANSPARENCY

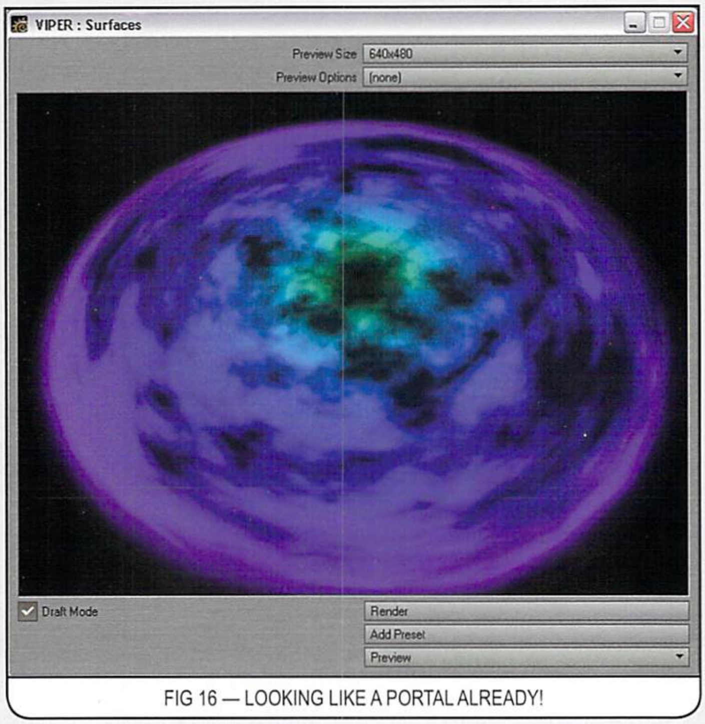

VIPER : Surfaces
Preview Size: 640x480
Preview Options: (none)
☑ Draft Mode

FIG 16 — LOOKING LIKE A PORTAL ALREADY!

Now we're pretty much finished with the modeling stage, so let's transfer over to Layout and start with the fun stuff! First off, we need to set the groundwork for alphering out the edges. At the moment, the outer and inner edges of the 'portal' disc have harsh edges, but we want to give the illusion of magical ethereal energy. Don't forget to save your model now.

Import the portal into Layout, and set up the camera so you can do a quick test render using VIPER. Right now, it just looks like a disc with a texture on it. There are a few ways we could give the edges a nice transparent look; one is to create a texture in Photoshop to use as a map, another way would be to use one of LightWave's built-in gradient features. Let's do that.

Open the Surface Editor, and click on the T button next to Transparency on our "Portal_Surface." Choose a gradient as the Layer type, and then in the input parameter choose Distance to Pivot Point. If you refer back to *(Fig 11)*, you'll notice that I set the top of the geometry at 0,0,0 in Modeler...this will put the pivot point right at the center of the portal, perfect for this gradient. If you haven't already done that, do so now.

So, what we want is for the center to be transparent, then fade in slowly and fade out again around the edges of the portal. Back in the Surface Editor, change the end of the gradient to 2. If you've been following along closely, you'll find your portal is 4 meters across, so the radius is 2 meters from the center. Put 100% Transparency at 0m and 2m, then 0% at 0.4m and 1.75m, and this should create a gradient like the one in *(Fig 15)*.

You should have something like in *(Fig 16)*. Not bad, so now how about giving it some movement. Unfortunately, up until LightWave [8] there was no inbuilt UV-Scrolling tool...But...there is a crazy little trick that can emulate it for us.

Go to the Image Editor and select the "FractalReflections.tga" file that we're using. So far, we've only used this texture as a temporary file. In Photoshop, or your favorite paint program, create a texture that is 512x512 in size and fill it with black. Save this image as "Black.tga." Back in Layout, replace "FractalReflections.tga" with this "Black.tga" file.

The reason we're doing this is because we're basically just going to project another image onto this one, so it is becoming like a canvas to paint onto and so doesn't need to have any colors. Since everything's gone black it won't matter if, under the Editing tab, we take the brightness down to −1 to make sure this image doesn't effect what we're about to project onto it. The portal is now completely black. Under the processing tab, add a filter from the list and choose "Filtered Image." This will allow us to basically project an image on top of the portal, while being able to keep the UVs, but the good thing it is animatable. Also, check the "Animate Filters" button, to allow animation on the texture. For some reason, LightWave [7.5] and earlier versions forgot to keep this checked when you saved the scene, so you need to recheck it when opening up the scene.

Double click on the Filtered image, select the Y-axis, and in the Texture tab, keep the projection to Planar, but also make sure the axis is on Y here as well. Now, for the image, choose anything you wish. I always say the best thing to do is to create custom textures, so follow along in the side-tab guide and make your own texture. Alternatively, just select "FractalReflections.tga" again.

Either way, whichever texture you decide to use here–either your created version of "PortalFractal_01.tga" or the generic "FractalReflections.tga"–the next step is to get it animating. Add a NULL object, and call it "UV_Animate." Back in the properties for the Textured Filter, make this NULL the Reference Object. Now, try moving about the NULL object a bit, and you should see that portal animates…In fact, keyframe it at 0,0,0 on Frame 0, then move it to Frame 100 and set it to 2,0,0. This should give a nice portal-like movement.

So hopefully by now you should have a nice looking portal that moves inwards. You could end here, but there's never really a need to stop working on something when you can modify and tweak it to make it better and better. There's always something more you can do to the effect. Let me show you a nice little trick that will bring this piece alive even more than it is now.

Save your model and scene, and then load it back up in Modeler. Change one of the viewports to the UV Texture display like in *(Fig 17)*. OK, here's the cool thing. What do you suppose would happen if we started to move about the UVs on the texture? Try it out. Move each UV about a bit, giving it a jittered look, but make sure not to touch the outer UVs *(Fig 18)*. Spend a bit of time doing this right, and you'll end up with some excellent results.

Save the model, and back in Layout do another test render. What you should find is that by jittering the UVs slightly, we've given the animation some randomness, so it doesn't appear to just move straight into the middle of the portal, but jitters about on the way. This is a really nice trick, and is especially useful for creating ethereal energy *(Fig 19)*.

I've shown you a nice little trick, but think of all the other cool things you could keep adding to this now that you have the groundwork…particles, lens flares, flashes, glows, a portal ring to hold the energy…the list is endless ...

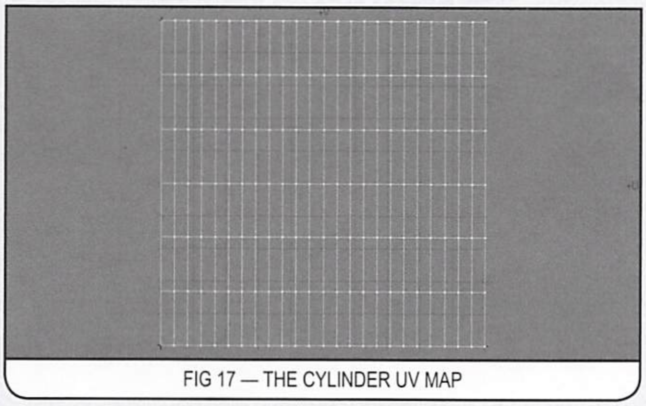


FIG 17 — THE CYLINDER UV MAP

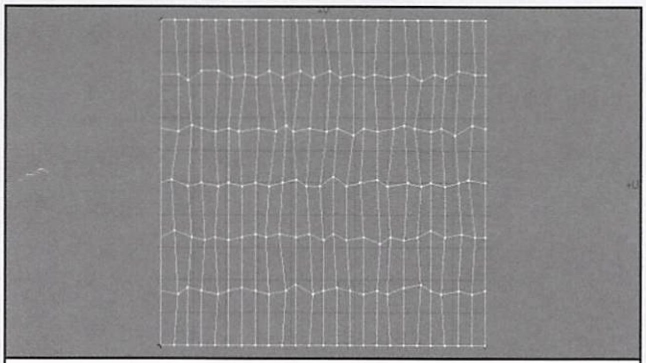
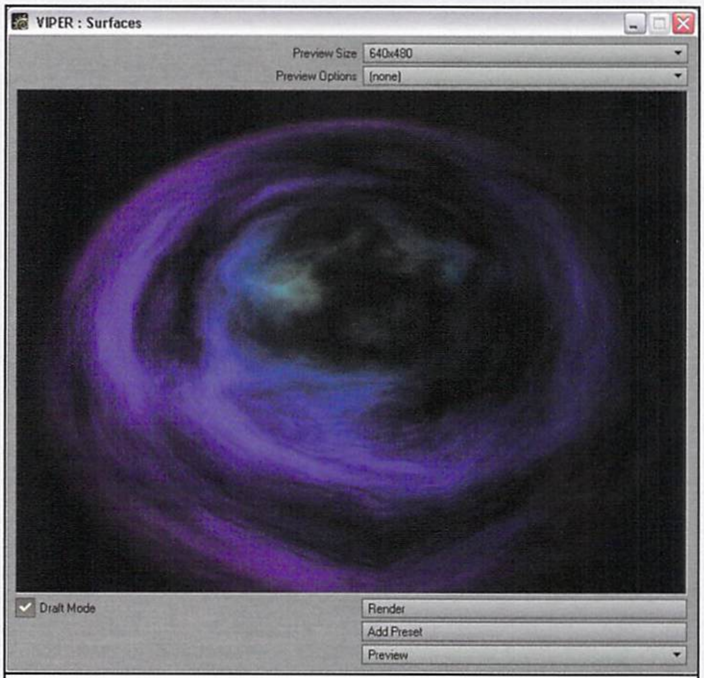


FIG 18 — JITTER THE UV POINTS ABOUT. BUT DON'T TOUCH THE EDGES



VIPER : Surfaces

Preview Size 640x480

Preview Options (none)

☑ Draft Mode

FIG 19 — THIS JITTERS ABOUT THE TEXTURE AS IT MOVES NOW

To finish off real quick, let's do just one more thing. In Modeler, select the portal, copy it and move the portal up slightly by about 50mm on Y. You should have two portals now, one slightly above the other. Put the second portal in another layer, and give it a new surface, such as "Portal_Surface2," and copy over all the attributes of the first surface to the new one with Surface Editor. In Layout, let's change a few attributes. First of all, in order to get this new surface to work independently of the first one, you'll need to also make a copy of that black texture and call it "black_512_2.tga."

CREATE YOUR OWN PORTAL TEXTURE

In Photoshop, create a new image at 512x512. From the Filters menu, choose the *Render>Clouds*. You'll see that this is instantly tileable; this is because we created it at 512 in size *(Fig A)*. There are so many ways to make textures, but I'll just show you here how you can create one very quickly and easily using Photoshop's built-in filters. Next, choose the *Distort>Glass* and play around with the settings if you wish. Here's the good part – run the *Render>Difference Clouds*, making sure to have black and white as your two primary colors. Do this four times, and you should have something like in *(Fig B)*. Save this now as "PortalFractal_01.tga."

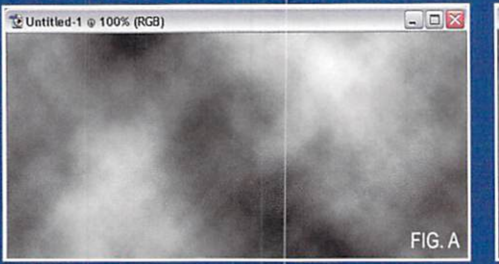


FIG. A

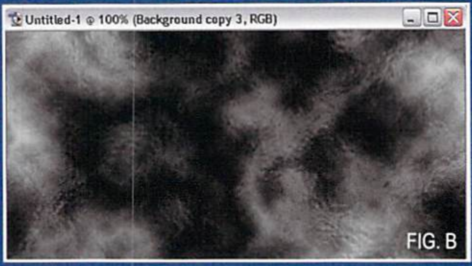


FIG. B

Not bad, but we can tweak it a bit more. Duplicate this layer onto another layer and run the *Render>Difference Clouds* one more time on this new layer. Change the blending mode of the new layer to Darken, then hit Ctrl+E to merge these two layers together. Bring up the contrast a little, and it should look like *(Fig C)*. Save this one as "PortalFractal_02.tga."

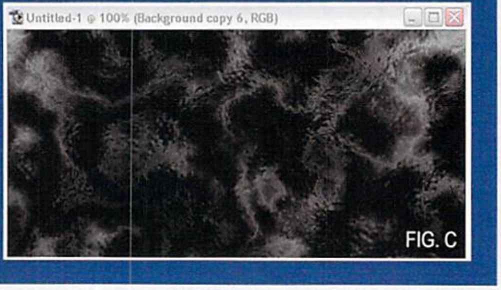
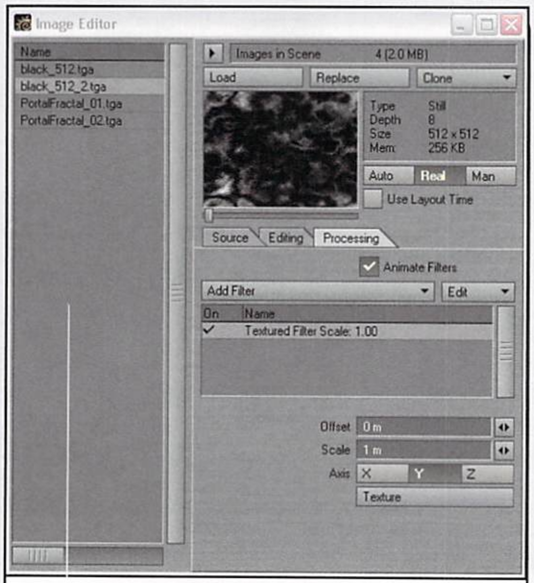


FIG. C



Name
black_512.tga
black_512_2.tga
PortalFractal_01.tga
PortalFractal_02.tga

Images in Scene 4 (2.0 MB)

Type Still
Depth 8
Size 512 x 512
Mem 256 KB

Auto Real Man

☐ Use Layout Time

☑ Animate Filters

On	Name
✓	Textured Filter Scale: 1.00

Offset 0 m
Scale 1 m
Axis X Y Z

FIG 20 — CREATE A SECOND SURFACE

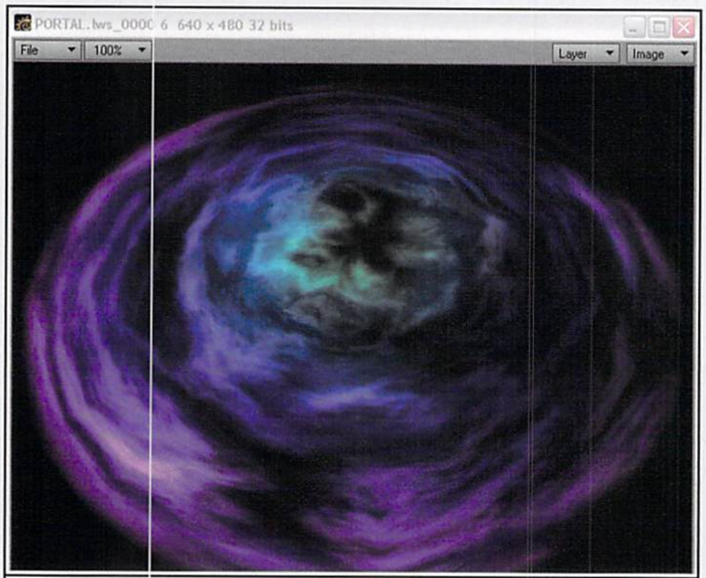


FIG 21 — USING YOUR OWN TEXTURES MAKES A BIG DIFFERENCE

Do the same Textured Filter trick to this new black texture, but this time use the "PortalFractal_02.tga" texture that you created in Photoshop using the side-panel tutorial *(Fig 20)*. What we're going to do is to add some more depth to the portal, so rather than it being this single flat texture, it will have much more substance to it when we use two separate and distinct images layered on top of each other.

Your object should now be made up of two layers, the first (the lower portal) and the second (the upper part). Select the second layer in Layout and in the Object Properties set the Object Dissolve to 50%. This will make it dissolve out, so we can see the lower portal underneath it. Also, to help in this, copy the Color channel into the Transparency channel. Make it additive with the transparency gradient and set it to about 30%. As a final last touch, you could add some fractal noise and ripples into the Color channel just to give it a bit more depth. By the end of it all, you'll probably have something similar to *(Fig 21)*.

I hope you've learned something from this tutorial, and if it's anything at all, then it is to just play around a bit and keep at it until you've got something you're really satisfied with. Remember, projects are never finished, only abandoned!

If you have questions or comments contact Shareef at: shareef@dx-digital.com

ALEXANDER SHAREEF HAS BEEN A PROFESSIONAL VISUAL EFFECTS ARTIST FOR THE LAST THREE YEARS, AND CURRENTLY WORKS AT BIOWARE CORP. IN CANADA AS A LEAD VISUAL EFFECTS ARTIST ON AN AS-YET-UNANNOUNCED PROJECT. HE WAS INSPIRED TO ENTER INTO 3D GRAPHICS AFTER WATCHING *BABYLON 5*, AND IS COMPLETELY SELF-TAUGHT. THOUGH HE HAS USED ALL THE MAJOR 3D APPLICATIONS, INCLUDING MAYA, 3DS MAX, AND HOUDINI, HE PREFERS THE ONE HE STARTED OUT WITH, LIGHTWAVE. HE MOVED TO CANADA FROM ENGLAND WITH HIS WIFE, AND IS FASCINATED WITH ANCIENT CULTURES SUCH AS ANCIENT EGYPT.

BY PETER RATNER

Setting up the Human Model for Animation

INTRODUCTION

If you have read other tutorials about rigging, then you are probably familiar with the usual disclaimer that says something like this, "This way of doing a rig is just one of many and is not meant to be the way of creating a rig." Yet, even though this has become a cliché, it is true. There are many ways of rigging and it is up to each animator to find the one they prefer.

The following tutorial will show you how to rig a biped character. It is a method that produces a rig that my students and I find easy to work with and does not require any programming skills to set it up. Piyush Patel from Digital Tutors teaches this system. Since Maya makes generous use of icons, it is a rig that fits right in with its interface. The skeleton is driven with foot, hand, and other icons representing parts of the human body. For example, when you want to move the right arm, grab the right hand icon and move it. If you want to lift the left leg, then move the left foot icon up.

Of course, before you can begin rigging you will need a human or human-like model. If you are unsure how to model one, then you can refer to my book, *Mastering 3D Animation, 2nd Edition*. It shows each step for building a subdivision or polygon model. Even though the book is non-software specific, you can adapt the method to smooth proxy modeling. Instead of spinning quads or faces, you can simply delete edges and reconnect the vertices to make new configurations.

Figure 1 shows an abbreviated series of steps for achieving a smooth proxy head. The advantage to starting with a box rather than extruding edges and gradually building a head from parts of it such as the eye area, nose, lips, and so on is that the box gives you an instant volume. One may ask why not start with a sphere since a human head appears to resemble that more than a cube. A sphere is vague and difficult to measure, and lacks distinct areas of association. The eye finds it difficult to focus on any one part of a sphere. However, a cube is easier to split into various components, less difficult to perceive in perspective, and appears to have the added dimension of weight. It is a much quicker method of modeling because instead of starting with details, you work with a general shape and gradually create more detailed areas. Proportions are easier to gauge when you already have the mass of the head. Unlike the edge extrude method; box modeling creates a head with a lot less polygons (faces).

One of the most invaluable contributors to 3D modeling has been Peter Levius. His site can be found at:

www.3d.sk/

It is by far the best place to find template photos of humans. This is an incredible site and definitely the Internet's most important resource for 3D modelers and texture artists. I hope more artists support this site so that it can continue to grow. My students and I find his site so valuable we are more than willing to promote it without making any money from it.

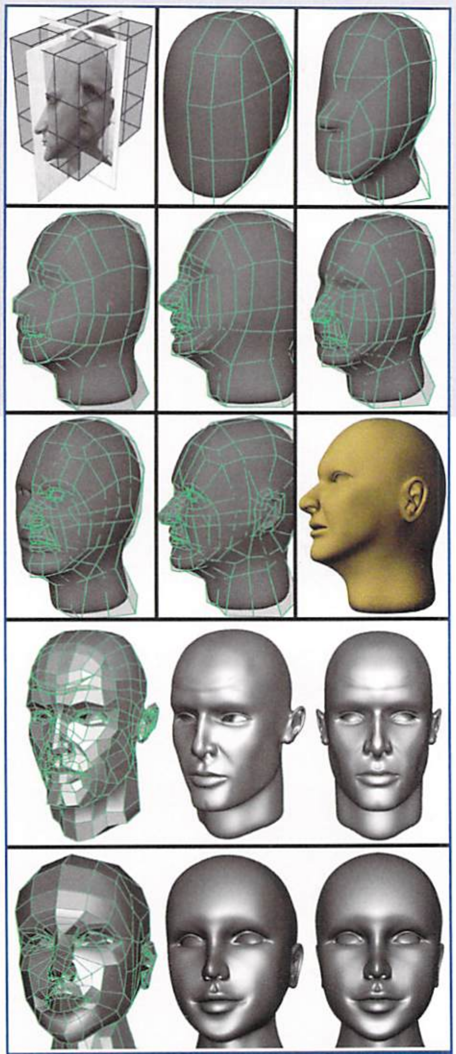


FIGURE 1 – ABBREVIATED MODELING STEPS FROM *MASTERING 3D ANIMATION, 2ND EDITION*

SETTING UP A SKELETON

In human anatomy, the skeleton carries the weight and serves as the foundation for the body. The muscles are attached to the skeleton with tendons. By contraction, muscles move the bones. The skeleton influences the form of the figure underneath the skin, fat, and muscles. In some parts of the body, the skeleton is always discernible on the surface, while in other areas, it only appears only during certain movements.

While 3D humans share some similar structural characteristics with real-life ones, the differences are much greater. The skeleton in animation is used mostly for movement and rarely adds structure to the body. Unless you are using a sophisticated muscular/skeletal system for animation, your character's bones will most likely look like simple geometric shapes. When these are rotated, they affect the polygons, splines, or NURBS mesh.

PREPARING FOR RIGGING A HUMAN MODEL

Each software package has its own methods for rigging a skeleton. Since Maya is considered the most popular and widely used 3D animation software in the movie industry, it behooves us to use its system of setting up a human for animation. If you are using something different than Maya, you may still find the following instructions useful by adapting them to your own method of working.

Before starting the actual rigging process, a few routines will be covered that are going to be utilized later. It is important to understand these concepts so that later when they are applied to a real skeleton setup they will not be too difficult to understand.

CREATING BONES WITH THE JOINT TOOL

Maya's tool for creating bones is called the Joint Tool. Joints are Maya's way of identifying the area where two or more bones are joined. The joints have a spherical look to them while the bones appear as elongated pyramid forms.

STEP 1

Select the Joint Tool for making bones by going to the Animation Menu Set on the top left of your screen. Select the menu item *Skeleton > Joint Tool* or select the Joint Tool from the Animation shelf. In the side view, starting at the top, click four times, moving the Joint Tool down in the same progression as seen in Figure 2.

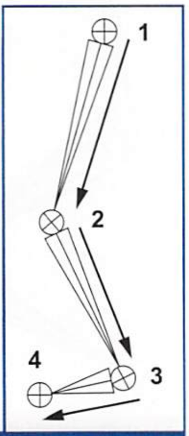


FIG. 2 – USING THE JOINT TOOL TO MAKE A SIMPLE LEG SKELETON

STEP 2

When you are finished clicking four times with the Joint Tool, press the Enter button to complete the simple leg skeleton. You can now adjust the size of each joint with the Move Tool. If the joints appear too large or small, go to the Display menu and select Joint Size.

TOGGLING LOCAL ROTATION AXES

Each joint has a local rotation axis that can be adjusted so that the bones rotate along the correct axis. For example, when you rotate the fingers into a closed fist, you would not want them to twist sideways along the wrong axis. To manually rotate a joint's local axis, do the following.

STEP 1

Select one of the joints on the leg skeleton. Select *Display > Component Display > Local Rotation Axes*. The joint should now display its x, y, and z orientation.

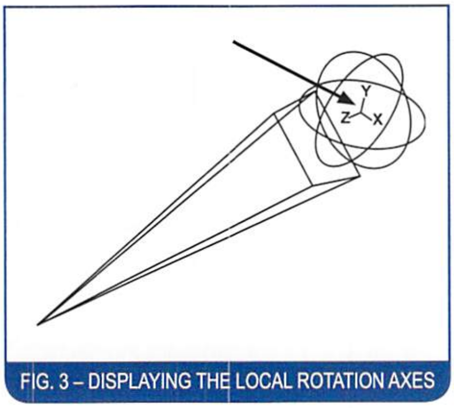
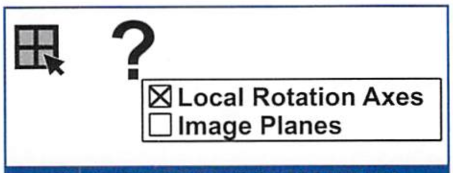


FIG. 3 – DISPLAYING THE LOCAL ROTATION AXES

Make a shelf icon for this command. Make sure the Animation shelf is in the foreground. Press the Shift and Control keys down while selecting *Display > Component Display > Local Rotation Axes.*

STEP 2

Click the Select by Component Type icon in the Status Line to turn on the component selection mode and also click the Miscellaneous (?) mask icon in the Status Line. Right click the Miscellaneous (?) mask icon and turn on Local Rotation Axes (Figure 4).



☒ Local Rotation Axes
☐ Image Planes

FIG. 4 – THE COMPONENT MODE AND THE MISCELLANEOUS MASK ICONS FOUND IN THE STATUS LINE

You can toggle back and forth between component mode and object mode by pressing F8.

STEP 3

In component mode, click on one of the three lines (y is green, z is blue, x is red). They are located inside each joint after selecting it and then going to *Display > Component Display > Local Rotation Axes.*

STEP 4

After selecting the local axis line, choose the Rotation Tool (press "e" on the keyboard). You can now rotate the three axes manually. For example, you may want to rotate the Y-axis of the joint so that it points in a straight line with the bone below it. When you are done orienting all the local rotation axes, press F8 to get out of component mode. Hide the local rotation axes of each joint by selecting the joint and pressing the shelf icon for *Display > Component Display > Local Rotation Axes.*

SELECTING A SKELETON'S HIERARCHY WITH A MEL SCRIPT

When you want to create movements such as fingers curling into a closed position simply by rotating one joint, you can use the following Mel script: "select -hi". By selecting all the connected bones and applying this script, you will be able to rotate the end joint, which in turn rotates the rest of the bones. You can also use this to bend a spine.

STEP 1

Open the Script Editor and type "select -hi". Highlight the two words that you typed and middle-click to drag the Mel script up to the Animation shelf.

STEP 2

Select the first joint located at the top of the leg bone chain. Choose the Rotate Tool. Rotate the leg bones in the side view. Notice that all the bones remain static while being rotated around the first joint's axis. Undo the rotation.

STEP 3

While the first joint is still selected, click the "select -hi" Mel script button on the Animation shelf. The rotate tool should now switch to the joint on the opposite end at the tip of the foot bone. Rotate this joint and watch how the other bones curl in. Later on when we curl the fingers closed for Set Driven Keys, we will use this Mel script. In your Shelf Editor, add an icon to this Mel script.

CREATING AN IK HANDLE ON A JOINT CHAIN

An IK handle is an animation tool for arranging joint chains. When you click on two joints, it creates a wire running through the joint chain. The IK handle that is created at the second joint can be moved, which in turn poses the entire joint chain. An IK solver automatically determines how to rotate all the joints in the joint chain.

STEP 1

Select the IK Handle Tool from the Animation shelf or go to *Skeleton > IK Handle Tool*. In the side view, click on the first joint at the top of the leg skeleton, then click again on the third joint located where the ankle would be. This should create a line between the two joints with an IK handle located at the ankle joint (Figure 5).

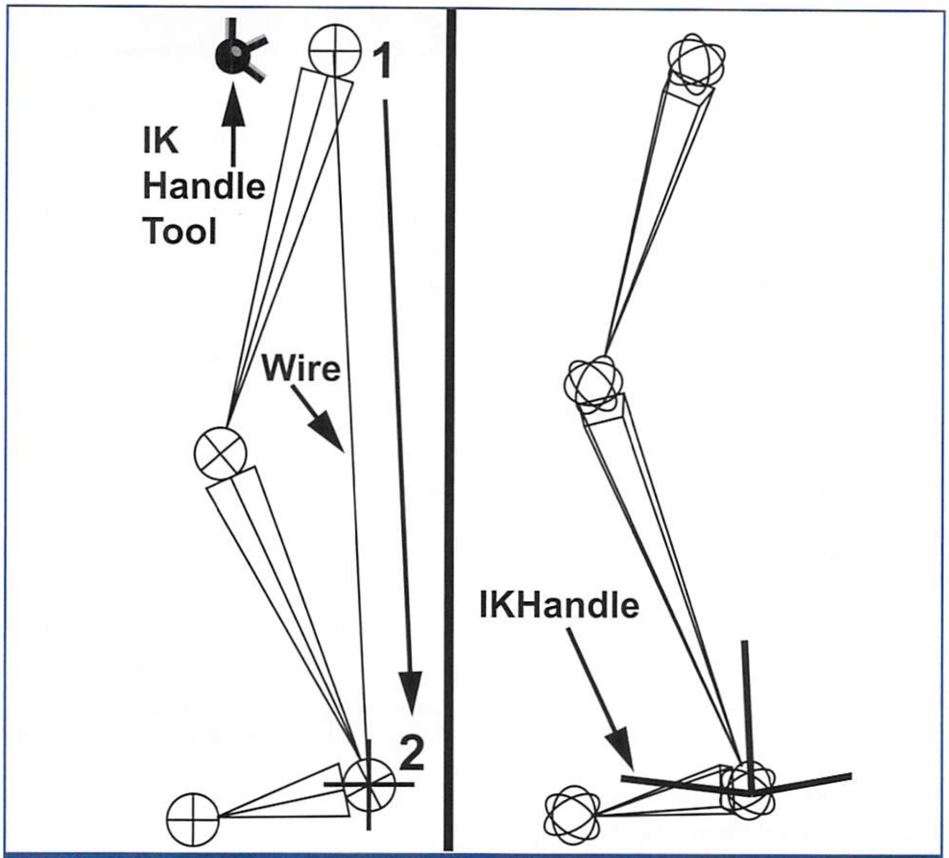


FIG. 5 – CLICKING ON THE FIRST AND THIRD JOINT WITH THE IK HANDLE TOOL CREATES A WIRE BETWEEN THE TWO JOINTS. THE IK HANDLE AT THE ANKLE JOINT IS USED TO MOVE THE LEG.

STEP 2

In the perspective window, select the IK Handle at the ankle joint and move it around with the Move Tool. Notice how the upper and lower leg bones resolve themselves according to how much the foot bone is raised.

ELIMINATING CHANNEL OPTIONS

The Channel Box contains quite a number of attributes such as translate, rotate, scale, and visibility. Some of these, such as scale and visibility, are not necessary for skeletal deformations and can be cleared out of the Channel Box, leaving room for you to add your own attributes.

STEP 1

Create an icon that can be used later to drive the IK Handle. Since we do not want the icon to render, it will be made from a curve. Go to *Create>NURBS Primitives>Circle*. Move the circle down and behind the foot (Figure 6).

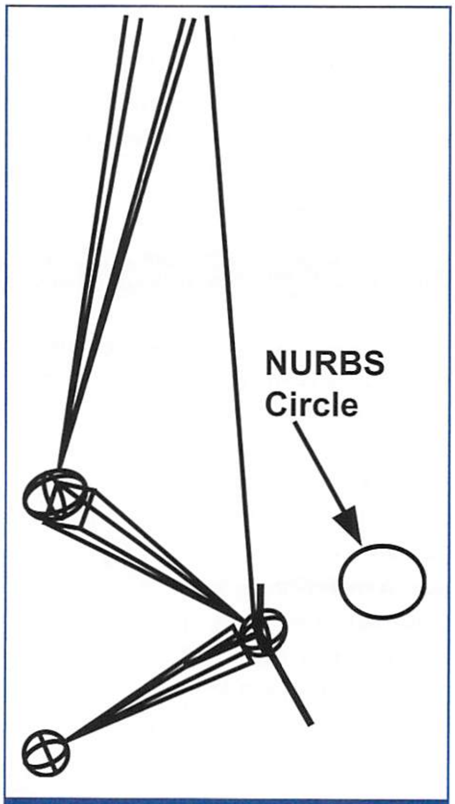


FIG. 6 – ADDING A NURBS CIRCLE THAT WILL DRIVE THE IK HANDLE

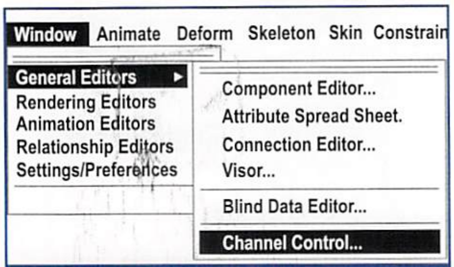

Window
Animate
Deform
Skeleton
Skin
Constrain
General Editors
Rendering Editors
Animation Editors
Relationship Editors
Settings/Preferences
Component Editor...
Attribute Spread Sheet.
Connection Editor...
Visor...
Blind Data Editor...
Channel Control...

FIG. 7 – OPENING THE CHANNEL CONTROL BOX IN ORDER TO ELIMINATE NONESSENTIAL ATTRIBUTES IN THE CHANNEL BOX

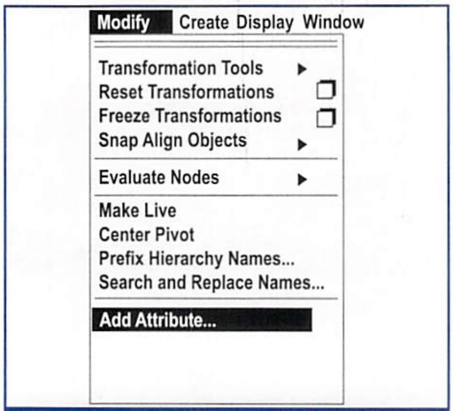

Modify
Create
Display
Window
Transformation Tools
Reset Transformations
Freeze Transformations
Snap Align Objects
Evaluate Nodes
Make Live
Center Pivot
Prefix Hierarchy Names...
Search and Replace Names...
Add Attribute...

FIG. 8 – OPENING THE ATTRIBUTE BOX FOR ADDING CERTAIN PROPERTIES TO THE CHANNEL BOX

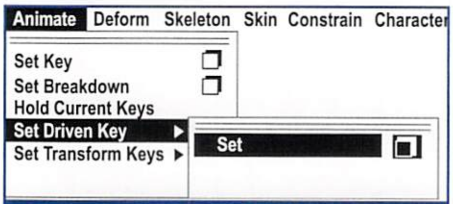

Animate
Deform
Skeleton
Skin
Constrain
Character
Set Key
Set Breakdown
Hold Current Keys
Set Driven Key
Set Transform Keys
Set

FIG. 9 – GOING TO THE SET DRIVEN KEYS OPTION BOX

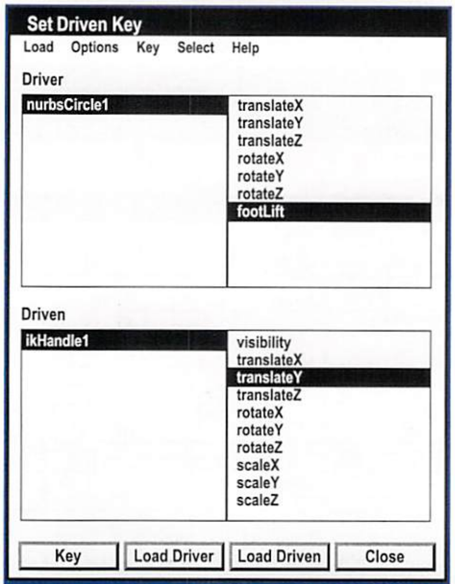

Set Driven Key
Load
Options
Key
Select
Help
Driver
nurbsCircle1
translateX
translateY
translateZ
rotateX
rotateY
rotateZ
footLift
Driven
ikHandle1
visibility
translateX
translateY
translateZ
rotateX
rotateY
rotateZ
scaleX
scaleY
scaleZ
Key
Load Driver
Load Driven
Close

FIG. 10 – THE SET DRIVEN KEYS OPTION BOX

STEP 2

Press "Control a" several times until you see the Channel Box on the top right of your screen. Notice all the attributes for the circle. We will now delete some of these.

STEP 3

Make sure the NURBS circle is selected and go to *Window > General Editors > Channel Control...*(Figure 7). On the left side of the Channel Control box, select only the Scale X, Y, and Z, plus the Visibility attributes. Click the Move button on the bottom of the box. You should now see that the Channel Box on the far right no longer exhibits those attributes. In the next part we are going to add our own attribute to the channel box.

ADDING ATTRIBUTES TO THE CHANNEL BOX

Adding the following attribute to the Channel Box will lead us into the next section on creating set driven keys.

STEP 1

While the NURBS circle is still selected, go to *Modify > Add Attribute...*(Figure 8).

STEP 2

In the Add Attribute box, under the New tab next to Attribute Name, type "FootLift" next to Attribute Name. Under the Numeric Attribute Properties, next to Minimum, type -5. Next to Maximum, type 10. Next to Default, type 0. Click OK. The Channel Box on the top far right should now show the new attribute "FootLift."

CREATING SET DRIVEN KEYS

Set Driven Keys are useful for making attributes drive the values of other attributes. In other words, when you change the value of one object (the driver) it will affect the behavior of another object (the driven).

STEP 1

Select the Animation Menu Set on the top left of your screen. Go to *Animate > Set Driven Keys > Set > Option Box* (Figure 9).

STEP 2

Notice that the Set Driven Keys box has a Driver at the top and a Driven on the bottom. The Driver is going to be the NURBS circle and the Driven is the IK Handle. Select the NURBS circle and click the "Load Driver" button on the bottom of the Set Driven Key box. The attributes from the Channel Box are now listed on the right next to the name of the NURBS circle. Select the attribute called "FootLift."

STEP 3

Select the IK Handle located at the ankle. In the Set Driven Key box click the "Load Driven" button. The IK Handle should now be listed under the Driven heading with its attributes on the right. In the Set Driven Key box, select only the "translateY" attribute next to the IK Handle (Figure 10).

STEP 4

In the Set Driven Key box, press the "Key" button on the bottom left. Select the NURBS circle and in its Channel Box on the top left of your screen, next to FootLift, type 10 and press Enter. The NURBS circle and the IK Handle can be selected within the Set Driven Keys box by clicking on their names. Select the IK Handle and move it straight up to bend the leg. Press the "Key" button on the bottom left of the Set Driven Key box. Close the Set Driven Key box.

STEP 5

Select the NURBS circle and in its Channel Box, highlight the name "FootLift." Click with your middle mouse button in one of the view windows and drag left and right. Notice that as you hold down the middle mouse button and drag back and forth, the foot raises and lowers according to the changing values of the NURBS circle in the Channel Box next to the highlighted attribute "FootLift." This is a simplified version of using Set Driven Keys. Later, when we rig a character, we will use this method to drive the feet and hands.

WEIGHTING AN IK SPLINE HANDLE

The spine of a character should have a lot of flexibility and be easy to control. This section shows how to set up an IK spine with an IK Spline Handle. All the control vertices of that IK Spline will be manipulated with a cluster deformer. Weights will then be assigned to the control vertices for easier handling of the spine.

STEP 1

Create a new scene and select the Joint Tool. In the side view starting at the bottom of the window and working up toward the top, click seven times to create six bones and seven joints. They should resemble a human spine. You can move the individual joints and also display them smaller by going to *Display > Joint Size.*

STEP 2

Go to *Skeleton > IK Spline Handle Tool > Options* (Figure 11). Under the IK Spline Handle Settings, next to Number of Spans, click 3. Using the IK Spline Handle Tool, click on the bottom joint and then the top joint of the chain. This will place an IK Handle at the top joint. In order to get a better view of just the spline and its control vertices you can hide the joints by going to your view window's Show menu and turning off Joints. You should now see a curving line that follows the path of the hidden joints.

STEP 3

Press F8 so that you can select the spline's vertices with component type. Draw a selection around all the components in your side view. You should now see the control vertices along the spline highlighted. Select the bottom three vertices and go to *Deform > Create Cluster.* This will give you a cluster handle that will move the IK Spline and its hidden skeleton. Select the next two vertices and go to Deform > Create Cluster. Select the top vertex and go to Deform > Create Cluster. You should now have three cluster handles along the spine that can control the way it bends (Figure 12). The next step will show how to get the base of the spine to remain stationary while the rest of the spine can be moved.

STEP 4

To add weights to the vertices, select all the control vertices and go to *Window > General Editors > Component Editor*...Figure 13 shows the three cluster handles listed under the Weighted Deformers tab. Under "cluster1" next to "cv[0]" type in 0 for its weight. This is the bottom control vertex, and with a weight of zero it will not move, thus giving stability to the spine. Next to "cv[1], type .3 for its weight. For "cv[2], type .6 for its weight. Under "cluster2" next to "cv[3]" type in .7. For "cv[4]" type in .8 for its weight. For "cluster3" next to "cv[5]" type in 1 for its weight. Close the Component Editor.

STEP 5

Press F8 to get out of component type and back to object type. Bring up the Perspective and Outliner windows. Rename the cluster handles. Name "cluster1Handle" to "SpineBase." Change "cluster2Handle" into "SpineMiddle." Rename "cluster3Handle" "SpineTop". Move the three cluster handles around to test the way the spine bends. You can always go back to *Windows > General Editors > Component Editor*...to change the weights of the vertices. Twisting the shoulders back and forth is done with the IK Handle at the top of the spine. Select it, and in the Channel Box, locate "Twist" and highlight it. Using your middle mouse button, drag back and forth to twist the spine.

STEP 6

Sometimes the cluster handles are difficult to select. To fix this problem, select one of the cluster handles and go to *Display > Component Display > Selection*

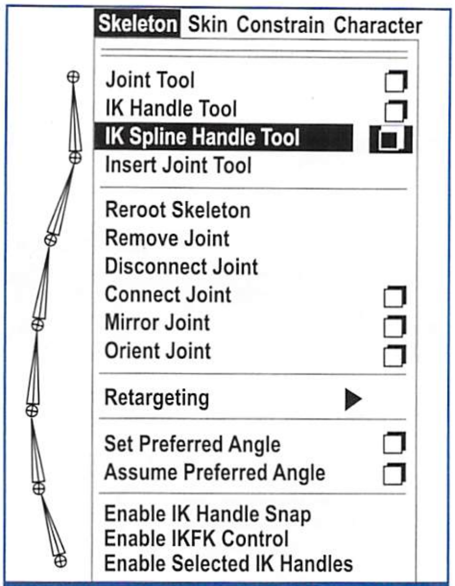

Skeleton
Skin
Constrain
Character
Joint Tool ☐
IK Handle Tool ☐
IK Spline Handle Tool ☐
Insert Joint Tool
Reroot Skeleton
Remove Joint
Disconnect Joint
Connect Joint ☐
Mirror Joint ☐
Orient Joint ☐
Retargeting ▶
Set Preferred Angle ☐
Assume Preferred Angle ☐
Enable IK Handle Snap
Enable IKFK Control
Enable Selected IK Handles

FIG. 11 – SELECTING THE IK SPLINE HANDLE TOOL OPTIONS. THE SPINE JOINTS ARE SHOWN ON THE LEFT

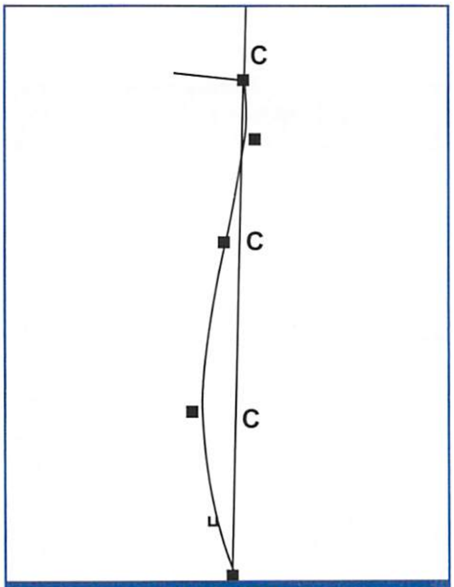
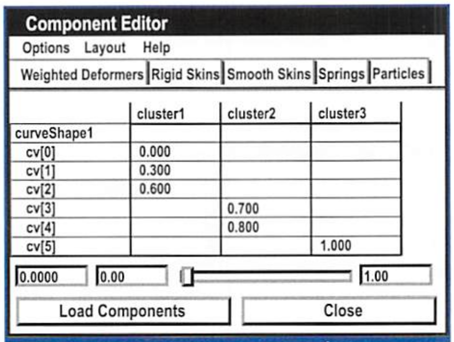


FIG. 12 – THE IK SPLINE CONTROL VERTICES AND THE THREE CLUSTER HANDLES


Component Editor
Options Layout Help
Weighted Deformers
Rigid Skins
Smooth Skins
Springs
Particles

	cluster1	cluster2	cluster3
curveShape1			
cv[0]	0.000		
cv[1]	0.300		
cv[2]	0.600		
cv[3]		0.700	
cv[4]		0.800	
cv[5]			1.000

0.0000 0.00 1.00

Load Components
Close

FIG. 13 – IN THE COMPONENT EDITOR, FOR EACH CONTROL VERTICES, USE THE SETTINGS SHOWN HERE TO GIVE EACH OF THEM A DIFFERENT WEIGHT.

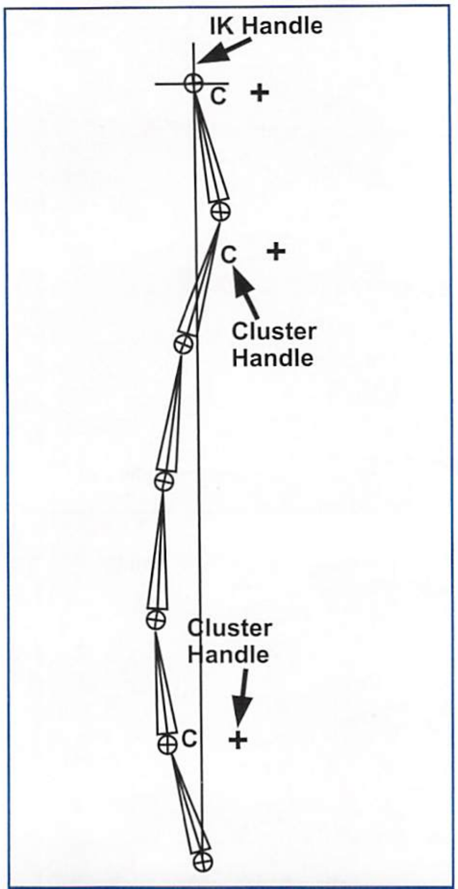

IK Handle
C
+
C
+
Cluster Handle
Cluster Handle
C
+

FIG. 14 – THE THREE SELECTION HANDLES (+ SIGNS) FOR MOVING EACH OF THE CLUSTER HANDLES.

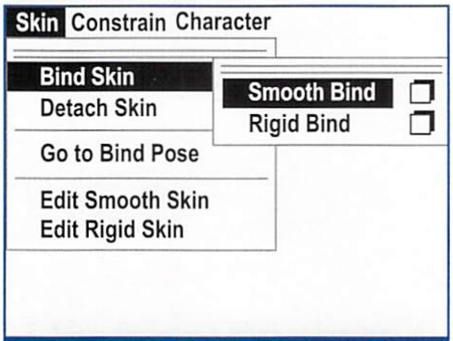

Skin Constrain Character
Bind Skin
Detach Skin
Go to Bind Pose
Edit Smooth Skin
Edit Rigid Skin
Smooth Bind ❐
Rigid Bind ❐

FIG. 16 – BINDING THE SKIN (MESH) TO THE SKELETON

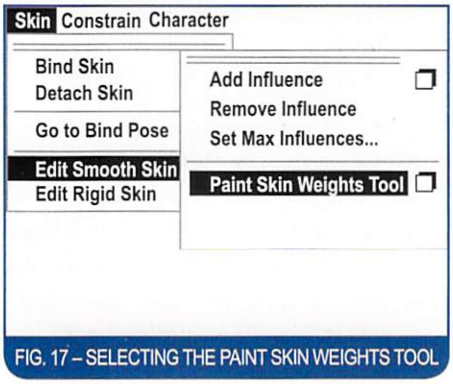
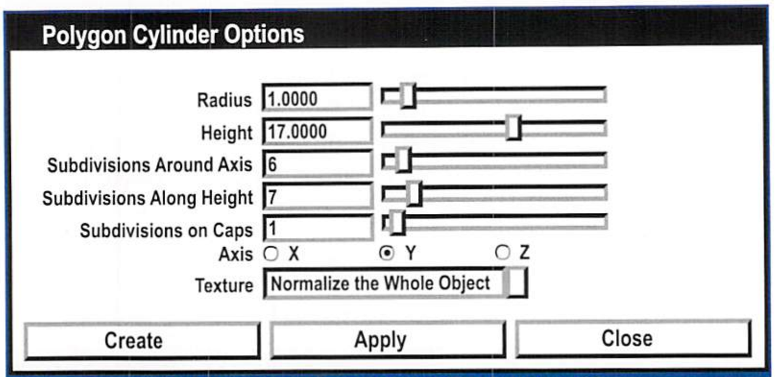

Skin Constrain Character
Bind Skin
Detach Skin
Go to Bind Pose
Edit Smooth Skin
Edit Rigid Skin
Add Influence ❐
Remove Influence
Set Max Influences...
Paint Skin Weights Tool ❐

FIG. 17 – SELECTING THE PAINT SKIN WEIGHTS TOOL


Polygon Cylinder Options

Radius 1.0000
Height 17.0000
Subdivisions Around Axis 6
Subdivisions Along Height 7
Subdivisions on Caps 1
Axis ○ X ● Y ○ Z
Texture Normalize the Whole Object

Create
Apply
Close

FIG. 15 – SETTINGS FOR THE POLYGON CYLINDER

Handles. This will put a small plus sign on the "C" cluster handle. Press F8 to go into component type and click the + sign (Selection Handles Component) at the top of the screen. Turn off all the other component types. This will make it easier to select the cluster + handle. Draw a selection around the selection handle. Move it behind the "C" cluster handle so it will be easy to find and select. Repeat these steps to make selection handles for the other two cluster handles (Figure 14).

BINDING SKIN (MESH) TO A SKELETON

The purpose of this exercise is to show how you can smooth bind a skeleton so that it influences its surrounding mesh.

STEP 1

Using your previously created spine setup, go to *Create > Polygons Primitives > Cylinder > Options*. In your Polygon Cylinder Options box, put in the settings seen in Figure 15 and click Create.

STEP 2

Turn on subdivision mode for the cylinder, or go to *Polygons > Smooth > Options* and set the Subdivision Levels to 3.

STEP 3

In wireframe view, select the root joint and Shift-select the cylinder. In the Animation Menu Set, go to *Skin > Bind Skin > Smooth Bind* (Figure 16). Using your Selection Handles and the Move Tool you can now deform the cylinder.

PAINTING WEIGHTS

Sometimes the mesh will not deform correctly at the joints. This can often be fixed by correcting the weights. These determine how much influence a particular bone exerts on the mesh. The Paint Skin Weights Tool is used to correct faulty weights.

STEP 1

Make sure that Smooth Bind on the mesh is still working. If moving Selection Handles does not affect the mesh, then go to *Skin > Bind Skin > Smooth Bind*. Sometimes switching back and forth between subdivision and low poly mode will detach the skin so that the skeleton no longer affects the mesh.

STEP 2

Select the cylinder and go to *Skin > Edit Smooth Skin > Paint Skin Weights Tool > Options* (Figure 17). The Paint Skin Weights Tool will now allow you to vary the weights on the cylinder.

When you select a joint under the Influence heading, it shows where and how much influence it has on the mesh. White means the most (100%) while black mean none or zero weight.

Press the "b" button to adjust the size of your brush. If you do not want to change the weight influence of a particular joint, then select it under the Influence heading and click the Toggle Hold Weights On Selected button. Now you select the joint above or below it and paint weights without changing the one that has "(Hold)" on it.

Under the Paint Weights heading next to Paint Operation you can use the following options:

REPLACE – This replaces the skin weight with the weight set for the brush.

ADD – You can increase the influence of nearby joints with this option.

SCALE – The power of far away joints can be diminished with this option.

SMOOTH – If you want to blend any rough transitions between weights, then use this option.

CLAMP – Allows you to hold certain settings for specified weights so that you never go below or above a particular minimum and maximum range.

FLOOD – This will apply all the brush settings to all the skin weights for the particular joint that is currently selected. For example, if you have the Replace value set to zero and you click Flood, it will put a zero (no weights) influence on that part of the mesh. The selected joint will no longer have any effect on any part of the cylinder.

STEP 3

When you are done, test the weights by moving the Selection Handles.

RIGID VS SMOOTH BINDING

There are two ways to bind a skeleton (Figure 18). You have already seen how to use the smooth bind option. Rigid binding differs from smooth binding because the control vertices of the mesh can only be assigned to one joint. For example, if you have a cylinder with two joints of equal length inside it and you select rigid binding, half the points on the mesh will be assigned solely to one joint, while the other half will only belong to the second joint. This results in deformations that appear crimped at the joints. Using a Paint Cluster Weights tool will resolve most of these problems (Figure 19). Rigid binding also allows one the option of applying lattices to control unwanted crimping at the joints (Figure 20).

On a smooth bind object, each CV (control vertex) can be assigned to multiple joints. The mesh around joints that are not being moved will also exhibit points around them being moved. This can result in some strange folding around joints that often includes a ballooning effect on the mesh. Setting the weights, such as is shown in Figure 17, will often fix these type of troubles. Rigid bound rigs perform faster than smooth bound ones.

IMPROVING DEFORMATIONS WITH FLEXORS

Continuing our work with the cylinder that has the spine joint setup, you will detach the skin from the smooth bound rig, bind it using Rigid Bind, and then edit the rigid skin with a Flexor.

STEP 1

Select the cylinder and go to *Skin > Detach Skin*. The joints will no longer deform the mesh.

STEP 2

Bind the skin by selecting the root joint first, and then Shift-select the cylinder. Go to *Skin > Bind Skin > Rigid Bind*. Test the rig by moving some of the selection handles and observe the crimping at the joints when the bones rotate more than 45 degrees. Undo all the movements of the selection handles so that the cylinder is in its upright position.

STEP 3

Go to *Skin > Edit Rigid Skin > Create Flexor...*(Figure 21). Accept the default settings in the Create Flexor option box.

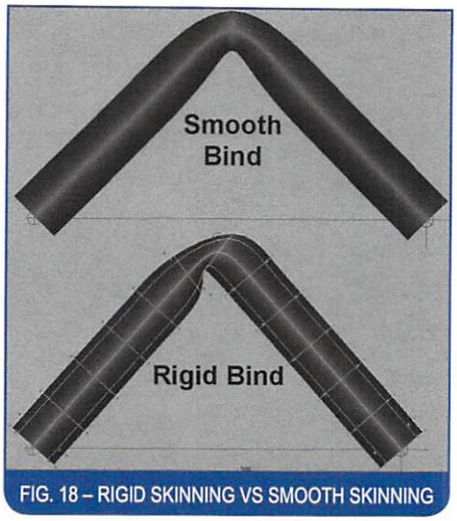


FIG. 18 – RIGID SKINNING VS SMOOTH SKINNING

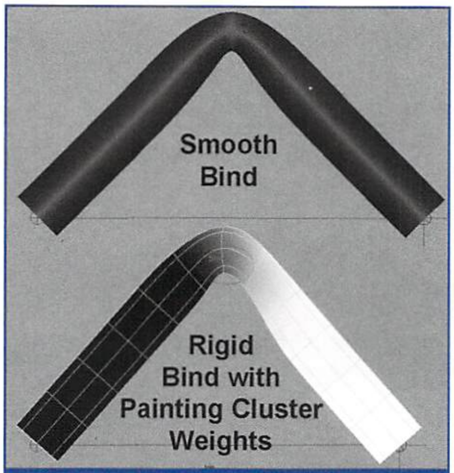


FIG. 19 – EDITING THE RIGID SKIN WITH THE PAINT CLUSTER WEIGHTS TOOL

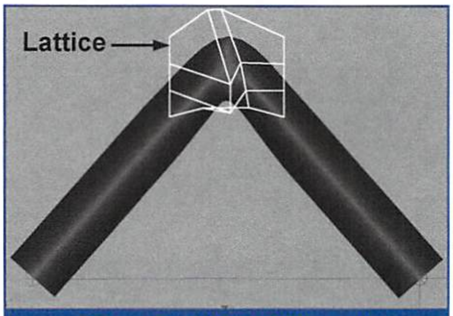


FIG. 20 – EDITING THE RIGID SKIN WITH A LATTICE

Test the effect of the lattice by moving the selection handle. If you still see too much crimping, then select the Lattice, right-click and select Lattice Point. Now select the lattice points in front of the area that is crimping. Use your Scale Tool to increase the distance between the lattice vertices on the Y-axis. Test

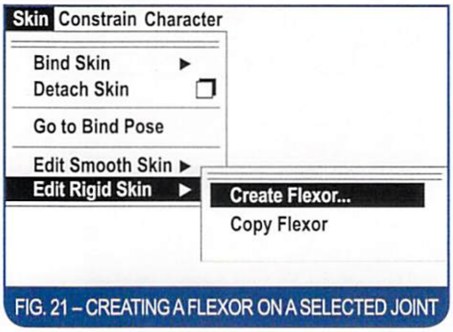

Skin Constrain Character
Bind Skin
Detach Skin
Go to Bind Pose
Edit Smooth Skin
Edit Rigid Skin
Create Flexor...
Copy Flexor

FIG. 21 – CREATING A FLEXOR ON A SELECTED JOINT

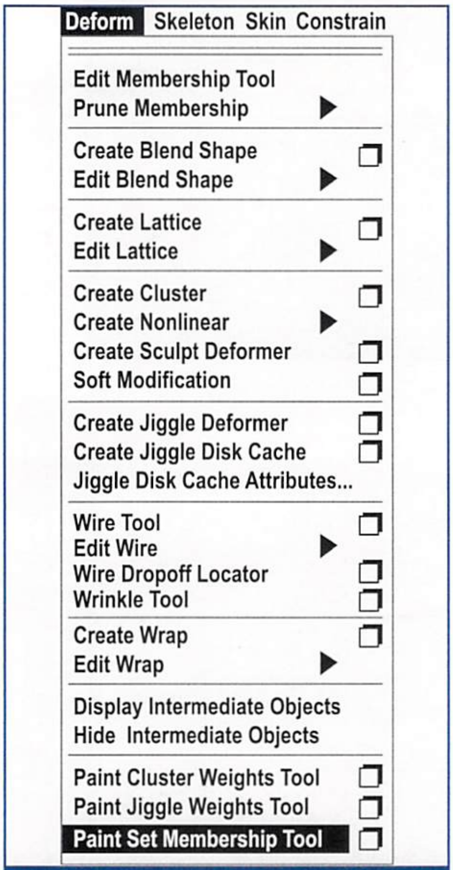

Deform Skeleton Skin Constrain
Edit Membership Tool
Prune Membership
Create Blend Shape
Edit Blend Shape
Create Lattice
Edit Lattice
Create Cluster
Create Nonlinear
Create Sculpt Deformer
Soft Modification
Create Jiggle Deformer
Create Jiggle Disk Cache
Jiggle Disk Cache Attributes...
Wire Tool
Edit Wire
Wire Dropoff Locator
Wrinkle Tool
Create Wrap
Edit Wrap
Display Intermediate Objects
Hide Intermediate Objects
Paint Cluster Weights Tool
Paint Jiggle Weights Tool
Paint Set Membership Tool

FIG. 22 – SELECTING THE PAINT SET MEMBERSHIP TOOL

the deformation again and keep scaling the lattice points on the Y-axis until you are satisfied. You can hide the Lattices by going to your view window's menu and selecting *Show > Deformers.*

PAINT SET MEMBERSHIP TOOL

Sometimes when you perform a rigid bind, certain vertices are assigned to the wrong bones. The Set Membership Tool allows you to reassign these vertices to the correct corresponding bones.

STEP 1

Select the cylinder and go to *Deform > Paint Set Membership Tool > Options.* In the Paint Set Membership Tool option box underneath Select Set to Modify, click on the joint sets until you see the vertices highlighted in your perspective view. After setting the size of your brush (b), you can Add, Transfer, or Remove to change the membership of individual points so that they correspond with the correct bones. You can find these under the title Paint Operations, and they do the following:

ADD – This removes the painted vertices from their current set and adds them to the selected set. Just select the joint set that you want to add points to and paint over the area that contains those vertices. They will then become part of the set that you had selected.

TRANSFER – The Transfer operation does the same thing as the Add operation: it removes the painted CVs or vertices from their current set and adds them to the selected set.

REMOVE – The Remove operation removes the painted CVs or vertices from the sets they belong to, so any deformers or joints do not influence the CVs vertices.

After experimenting with the Paint Set Membership Tool, move the selection handles to see if you get any improved deformations at the joints. If you see that your are still getting some crimping then leave the cylinder in its bent condition and go back to the Paint Set Membership Tool. Find the joint set at the deformed area. Add or Transfer the vertices, and see what happens. You should be able to fix the problem.

Now that you have had a chance to experiment with the various operations that will be used for rigging a character, the next part of this tutorial, in the next issue, will have you rig a biped model. During the process, if you find certain aspects confusing, then just review this beginning section titled *Preparing for Rigging a Human Model.*

PETER RATNER IS A PROFESSOR OF 3D COMPUTER ANIMATION AT JAMES MADISON UNIVERSITY. HE IS THE FOUNDER AND HEAD OF THE FIRST COMPUTER ANIMATION PROGRAM IN VIRGINIA. HIS PAINTINGS, ANIMATIONS, AND COMPUTER GRAPHICS HAVE BEEN DISPLAYED IN NUMEROUS NATIONAL AND INTERNATIONAL JURIED EXHIBITIONS. HE IS THE AUTHOR OF *3-D HUMAN MODELING AND ANIMATION, 1ST AND 2ND EDITIONS* (JOHN WILEY AND SONS) AND *MASTERING 3D ANIMATION, 1ST AND 2ND EDITIONS* (ALLWORTH PRESS). HE LIVES IN PENN LAIRD, VIRGINIA.

BY ANDREA CARVEY

LINKING OBJECTS TO PARTICLES
PART 1 - FX LINKER

Once in a while, something that is inherently very difficult and time consuming is made fantastically simple with the right tools. FX Link and FX Linker are two tools that are easy to use with little fiddling needed, yet have many parameters that can be adjusted to suit your needs. And they produce some impressive and fun animations to boot!

Basically, you set up a particle emitter and then "link" an object to the motion of one of the particles. The difference between FX Link and FX Linker is that with FX Linker, an object is cloned and each clone is linked to a different particle. The rotation and spin of the clones can be adjusted to give a natural look. However, you can't specify which particle an object is linked to, whereas with FX Link you can. As always, there are pros and cons for each of the tools, and which one you use will depend on what you need to do. In this issue, we will discuss the use of FX Linker and next issue we will look at FX Link.

SET UP THE HV EMITTER

BEGIN WITH CLEARED SCENE
ADD HV EMITTER

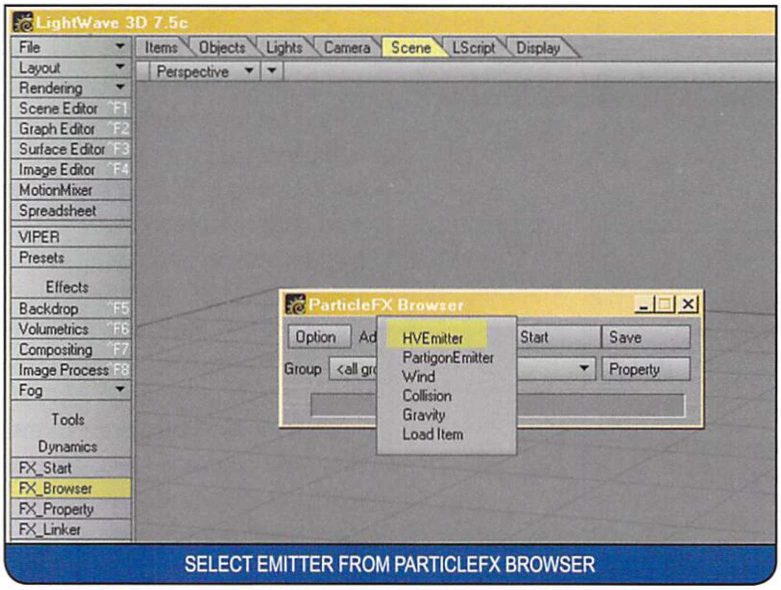
1. **Open the ParticleFX Browser** - Scene / FX_Browser (in the Tools/Dynamic section)

2. **Select "HV Emitter"** from the "Add" drop-down menu (It must be clicked twice, once to access the drop-down menu, another time to select the HV Emitter.)

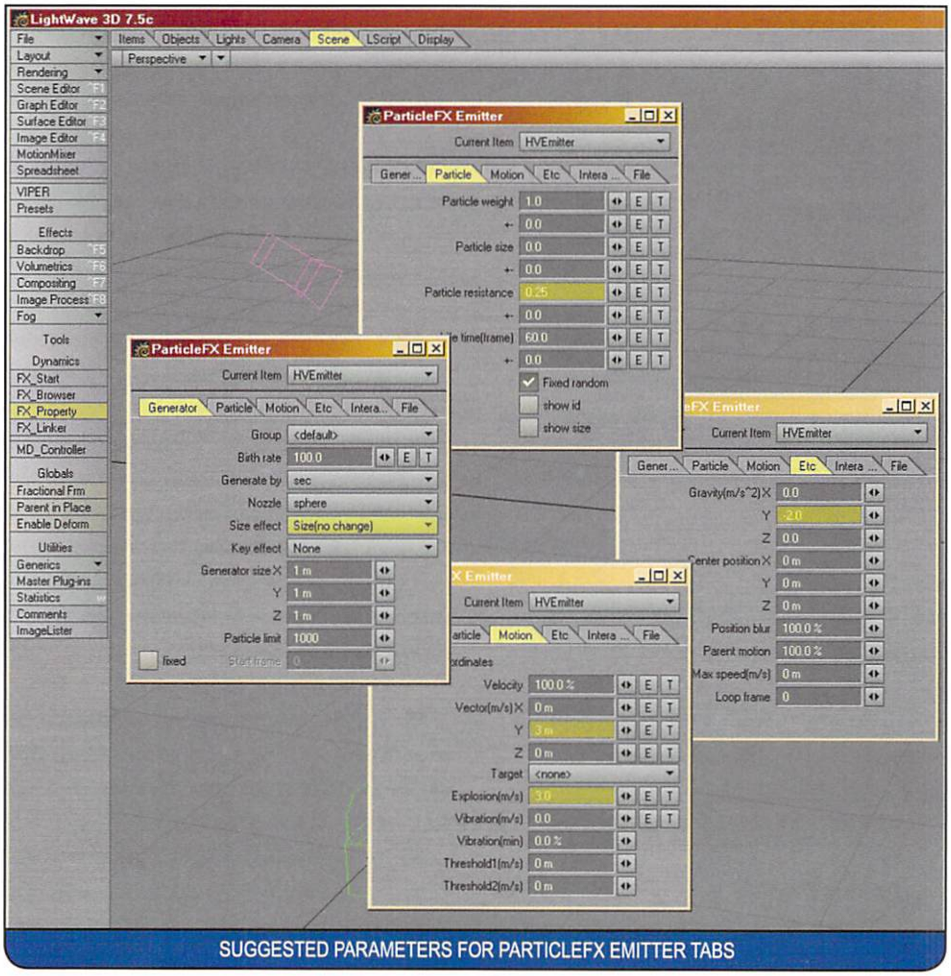
3. **Close** the ParticleFX Browser

A NOTE ABOUT STEP 2

Either the HV Emitter or the Partigon Emitter will work, but Partigons that are not assigned to an object will render as tiny sparks, whereas particles from the HV Emitter will not be visible. If you want the sparks, no problem, but by changing the Surface Transparency of the Partigon Emitter to 100%, they become invisible.


LightWave 3D 7.5c
File
Layout
Rendering
Scene Editor
Graph Editor
Surface Editor
Image Editor
MotionMixer
Spreadsheet
VIPER
Presets
Effects
Backdrop
Volumetrics
Compositing
Image Process
Fog
Tools
Dynamics
FX_Start
FX_Browser
FX_Property
FX_Linker
Items
Objects
Lights
Camera
Scene
LScript
Display
Perspective
ParticleFX Browser
Option
Ad
Start
Save
Group
<all gr
Property
HVEmitter
PartigonEmitter
Wind
Collision
Gravity
Load Item

SELECT EMITTER FROM PARTICLEFX BROWSER


LightWave 3D 7.5c
File
Layout
Rendering
Scene Editor
Graph Editor
Surface Editor
Image Editor
MotionMixer
Spreadsheet
VIPER
Presets
Effects
Backdrop
Volumetrics
Compositing
Image Process
Fog
Tools
Dynamics
FX_Start
FX_Browser
FX_Property
FX_Linker
MD_Controller
Globals
Fractional Frm
Parent in Place
Enable Deform
Utilities
Generics
Master Plug-ins
Statistics
Comments
ImageLister
Items
Objects
Lights
Camera
Scene
LScript
Display
Perspective
ParticleFX Emitter

Current Item HVEmitter

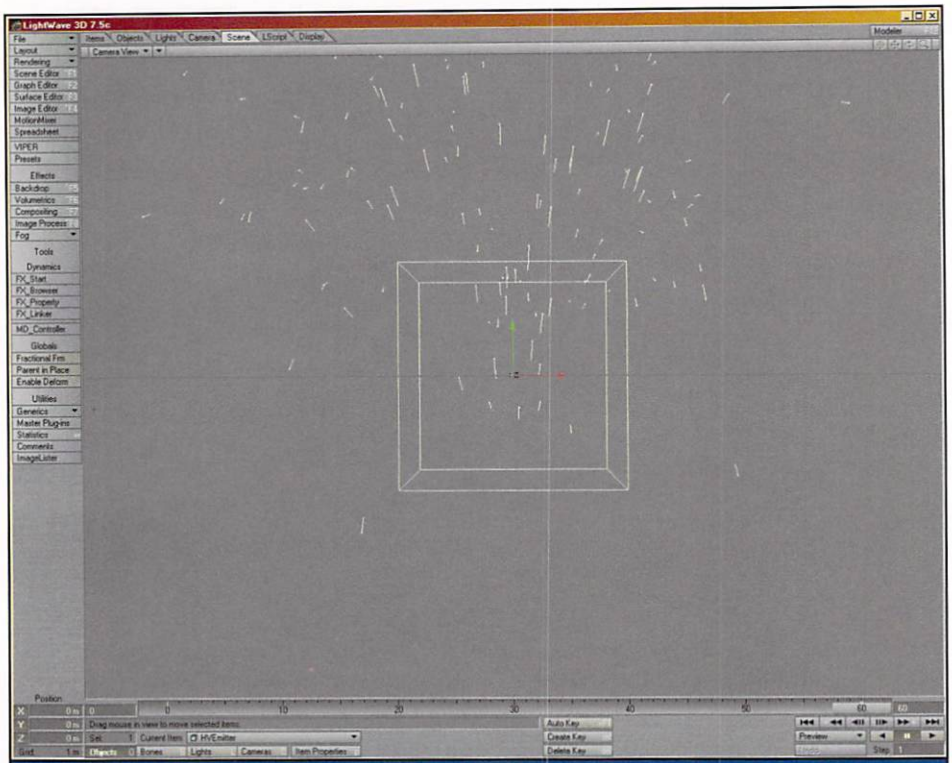
Generator
Group <default>
Birth rate 100.0
Generate by sec
Nozzle sphere
Size effect Size(no change)
Key effect None
Generator size X 1 m
Y 1 m
Z 1 m
Particle limit 1000
☐ fixed
Start frame 0

Particle
Particle weight 1.0
+- 0.0
Particle size 0.0
+- 0.0
Particle resistance 0.25
+- 0.0
e time(frame) 60.0
+- 0.0
☑ Fixed random
☐ show id
☐ show size

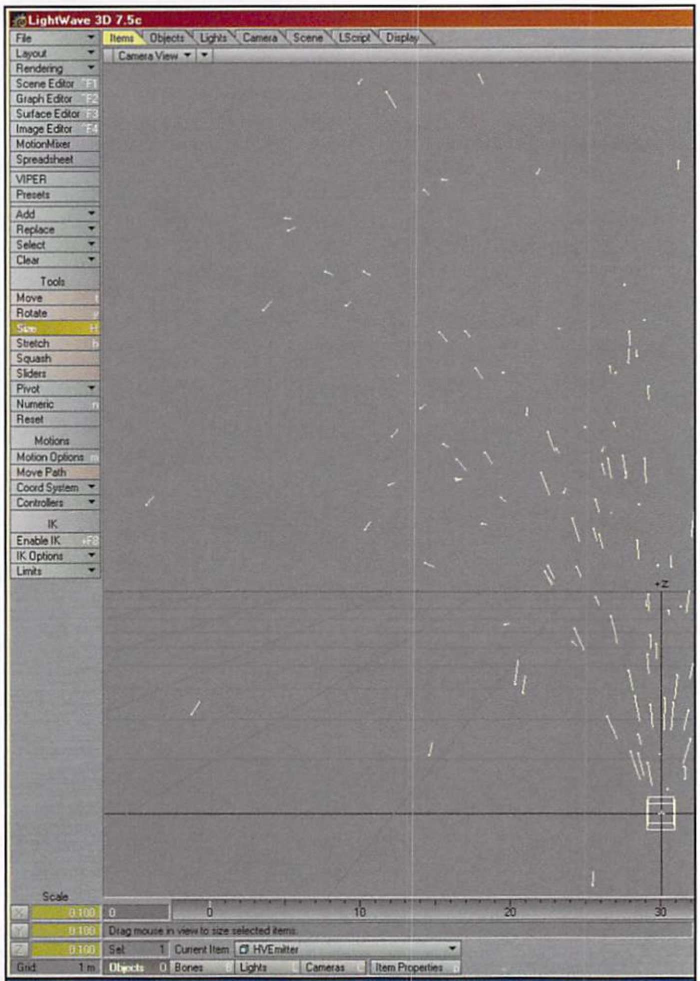
Motion
Velocity 100.0 %
Vector(m/s) X 0 m
Y 3 m
Z 0 m
Target <none>
Explosion(m/s) 3.0
Vibration(m/s) 0.0
Vibration(min) 0.0 %
Threshold1(m/s) 0 m
Threshold2(m/s) 0 m

Etc
Gravity(m/s^2) X 0.0
Y -2.0
Z 0.0
Center position X 0 m
Y 0 m
Z 0 m
Position blur 100.0 %
Parent motion 100.0 %
Max speed(m/s) 0 m
Loop frame 0

SUGGESTED PARAMETERS FOR PARTICLEFX EMITTER TABS


LightWave 3D 7.5c
Items
Objects
Lights
Camera
Scene
LScript
Display
Camera View
Modeler

PARTICLES EMITTING FROM EMITTER OBJECT


LightWave 3D 7.5c
File
Layout
Rendering
Scene Editor
Graph Editor
Surface Editor
Image Editor
MotionMixer
Spreadsheet
VIPER
Presets
Add
Replace
Select
Clear
Tools
Move
Rotate
Size
Stretch
Squash
Sliders
Pivot
Numeric
Reset
Motions
Motion Options
Move Path
Coord System
Controllers
IK
Enable IK
IK Options
Limits
Items
Objects
Lights
Camera
Scene
LScript
Display
Camera View
Scale
Drag mouse in view to size selected items.
Current Item
HVEmitter
Grid
Bones
Lights
Cameras
Item Properties

SIZED EMITTER OBJECT

CREATE PARTICLE MOTION

4. Select the HV Emitter

5. Open the "ParticleFX Emitter" window - Scene / FX Property

6. Change the following default settings:

Generator Tab– **Size Effect** = **Size** (no change)
Particle Tab– Resistance = .25
Motion Tab– Vector (m/s) y = 3 m
Explosion (m/s) = 3.0
Etc. Tab– Gravity (m/s2) y = -1.0

7. Close the "ParticleFX Emitter" window

Particles should emit over 60 frames.

NOTES ABOUT STEP 5

GENERATOR TAB

One weird quirk occurs occasionally. If one of the other "Size Effect" parameters is selected, sometimes only one particle apparently is emitted, and makes it seem like nothing is happening.

PARTICLE TAB

Under the Particle Tab of the ParticleFX Emitter window, there is a parameter called "Particle Lifetime." This determines how long (in frames) a particle "lives" or is visible and moves. To prevent the object that is linked to a particle from stopping in mid-air, make sure that the Particle Lifetime is equal to or longer than the animation.

SET UP SCENE

8. Load Object
Use LightWave / Objects / Food / Apple (or Banana).

9. Move Camera
Suggestion: y = 1.25 m; z = - 5m
(Create keyframe at frame "0").

10. Size HV Emitter
Select HV Emitter.
Change "Scale" - Suggestion x, y, z, = .1 (Items / Size). (Create keyframe at frame "0".)

11. Save Scene
(There is no Undo for FX Linker, so to "undo" the linking procedure, you have to reload the scene.)

LINK PARTICLES TO OBJECTS

12. Select Object (apple or banana)

13. Open the ParticleFX Linker window (Scene / FX Linker).

15. Enter parameters as indicated in Figure at Right.

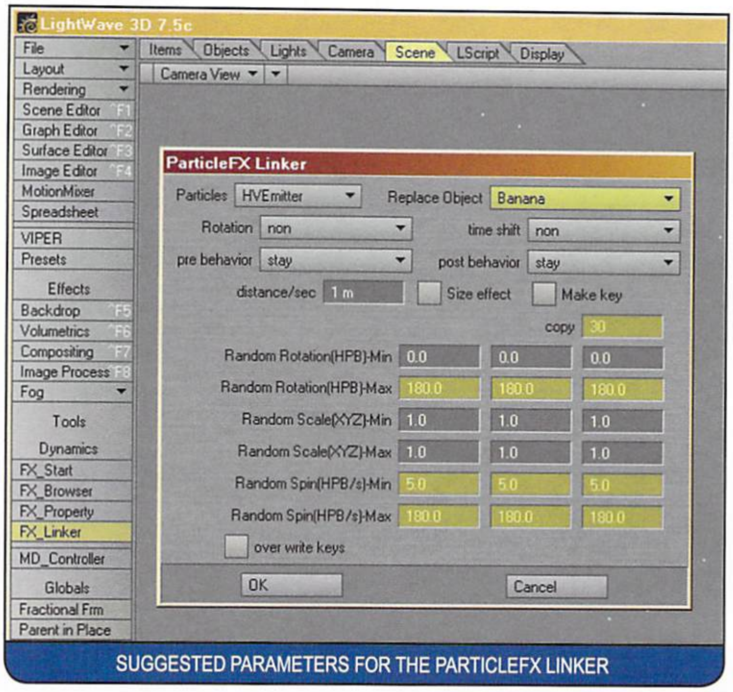
16. Click "OK."

Now when you run the animation, 31 apples (or bananas) should all fly up in the air like they were kicked by a donkey (see picture Lower Right). Pretty darn neat!

If you want to change the behavior of the objects, you have to reload the scene that you saved prior to the linking procedure and enter different parameters for the particle emitter (Step 6) and/or the FX Linker (Step 15).

So, it's just that simple! You can play around with the parameters to get different particle effects and link any objects you want. Just think — fluttering leaves, gold coins, flying debris. I'm sure you'll come up with something.

BRAD AND ANDREA CARVEY HAVE BEEN DOING COMPUTER ANIMATIONS FOR A LONG TIME. IN 1969 BRAD USED AN ANALOG COMPUTER, WHICH WAS THE SIZE OF A CAR, TO PRODUCE HIS FIRST COMPUTER ANIMATION. ANDREA, AN ARCHEOLOGIST, PREFERS TO DO SCIENTIFIC ANIMATIONS; HER CREDITS INCLUDE PROGRAMS LIKE DISCOVERY CHANNEL'S *UNDERSTANDING CARS*. BRAD IS AN ELECTRICAL ENGINEER AND AN EMMY AWARD-WINNING MEMBER OF THE VIDEO TOASTER DEVELOPMENT TEAM. HE PREFERS TO DO FEATURE FILM WORK. HIS CREDITS INCLUDE FILMS LIKE *MEN IN BLACK*, *STUART LITTLE*, *BLACK HAWK DOWN*, *KATE & LEOPOLD* AND *MASTER OF DISGUISE*.

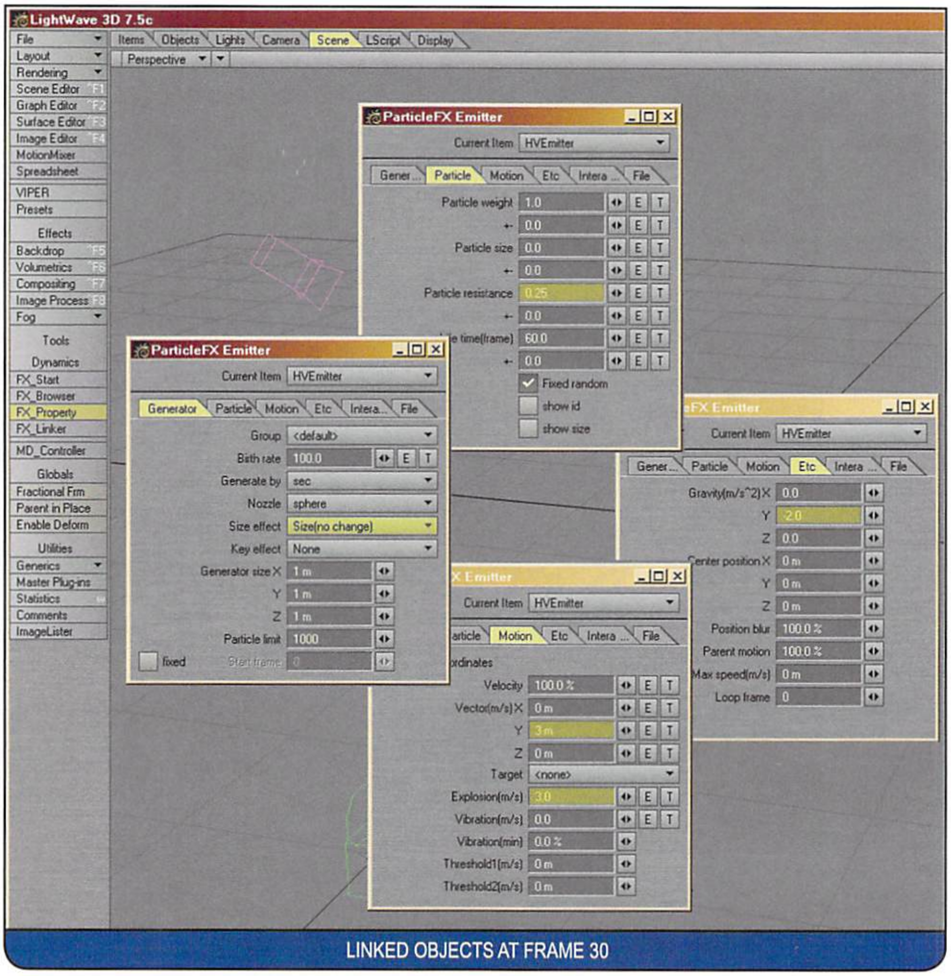

LightWave 3D 7.5c
File
Layout
Rendering
Scene Editor
Graph Editor
Surface Editor
Image Editor
MotionMixer
Spreadsheet
VIPER
Presets
Effects
Backdrop
Volumetrics
Compositing
Image Process
Fog
Tools
Dynamics
FX_Start
FX_Browser
FX_Property
FX_Linker
MD_Controller
Globals
Fractional Frm
Parent in Place
Items
Objects
Lights
Camera
Scene
LScript
Display
Camera View
ParticleFX Linker
OK
Cancel

Particles HVEmitter
Replace Object Banana
Rotation non
time shift non
pre behavior stay
post behavior stay
distance/sec 1 m
☐ Size effect
☐ Make key
copy 30

Random Rotation(HPB)-Min	0.0	0.0	0.0
Random Rotation(HPB)-Max	180.0	180.0	180.0
Random Scale(XYZ)-Min	1.0	1.0	1.0
Random Scale(XYZ)-Max	1.0	1.0	1.0
Random Spin(HPB/s)-Min	5.0	5.0	5.0
Random Spin(HPB/s)-Max	180.0	180.0	180.0

☐ over write keys

SUGGESTED PARAMETERS FOR THE PARTICLEFX LINKER


LightWave 3D 7.5c
File
Layout
Rendering
Scene Editor
Graph Editor
Surface Editor
Image Editor
MotionMixer
Spreadsheet
VIPER
Presets
Effects
Backdrop
Volumetrics
Compositing
Image Process
Fog
Tools
Dynamics
FX_Start
FX_Browser
FX_Property
FX_Linker
MD_Controller
Globals
Fractional Frm
Parent in Place
Enable Deform
Utilities
Generics
Master Plug-ins
Statistics
Comments
ImageLister
Items
Objects
Lights
Camera
Scene
LScript
Display
Perspective
ParticleFX Emitter
Generator
Particle
Motion
Etc
Intera...
File

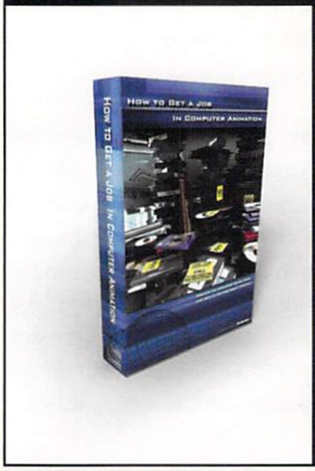
ParticleFX Emitter — Generator
Current Item HVEmitter
Group <default>
Birth rate 100.0
Generate by sec
Nozzle sphere
Size effect Size(no change)
Key effect None
Generator size X 1 m
Y 1 m
Z 1 m
Particle limit 1000
☐ fixed
Start frame 0

ParticleFX Emitter — Particle
Current Item HVEmitter
Particle weight 1.0
+- 0.0
Particle size 0.0
+- 0.0
Particle resistance 0.25
+- 0.0
Life time(frame) 60.0
+- 0.0
☑ Fixed random
☐ show id
☐ show size

ParticleFX Emitter — Etc
Current Item HVEmitter
Gravity(m/s^2) X 0.0
Y -2.0
Z 0.0
Center position X 0 m
Y 0 m
Z 0 m
Position blur 100.0 %
Parent motion 100.0 %
Max speed(m/s) 0 m
Loop frame 0

ParticleFX Emitter — Motion
Current Item HVEmitter
ordinates
Velocity 100.0 %
Vector(m/s) X 0 m
Y 3 m
Z 0 m
Target <none>
Explosion(m/s) 3.0
Vibration(m/s) 0.0
Vibration(min) 0.0 %
Threshold1(m/s) 0 m
Threshold2(m/s) 0 m

LINKED OBJECTS AT FRAME 30


How to Get a Job in Computer Animation

PROBLEM:

Tired of looking for a computer animation job?

SOLUTION:

How to Get a Job in Computer Animation.

"The best book for any CG artist looking for work."

- Worlds largest public listing of production & game companies, ever.
- Tips for making the perfect demo reel.
- How much you should get paid.
- Interviewing for animation jobs.
- How to avoid losing your job.
- Guide to successful freelancing.
- Writing a résumé and cover letter.
- Picking an animation school.
- Choosing CG hardware and software.
- And much, much more...

www.GetA3DJob.com

www.EdHarriss.com

VT[4]™

- Custom VT[4]™ Systems
- Professional Installation and Training

Digitek Systems, Inc.

LIGHTWAVE® [8]

- LightWave 3D® Sales and Support

1009 Dresser Ct. • Raleigh, NC 27609
919-790-5488 • www.digiteksys.com
Authorized NewTek Reseller and Service Provider

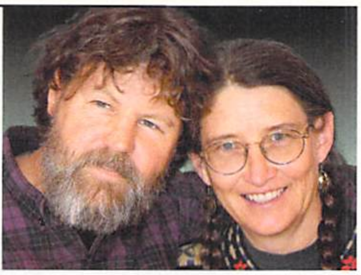
LightWave and LightWave 3D are registered trademarks of NewTek, Inc. VT[4] is a trademark of NewTek, Inc.

1980
2005
PINNACLE SYSTEMS
MIRAGE
ANARK
DIGITAL FUSION
SERIOUS MAGIC
HUGE SYSTEMS
Medéa
LEITCH

Celebrating Our 25th Anniversary

T. S. Computers 11300 Hartland St. N. Hollywood CA 91605
818-760-4445 www.vgn.com

TRACKING THE LEARNING CURVE



BY ANDREA CARVEY

Most of us spend a good deal of our lives learning stuff. As human beings, we are really good at learning stuff. In fact, when there isn't enough stuff for us to learn, we start making stuff up (baseball statistics, the IRS tax code, or crop circles, for instance). Unfortunately, that doesn't mean learning is always easy, or fun. As luck would have it, the stuff we need to learn is usually the hardest and least fun, whereas stuff like trivia sticks in our minds like cat hair on ice cream.

The kind of information that we really need requires work and time, and this goes double for the 3D world. In other words, there is a learning curve involved in gaining information. I have experienced a number of different learning curves in my travels through 3D-land, many of which are familiar to most people.

First is the classic linear "curve." Technically, it is a slope rather than a true curve. It has a beginning, a rigid, focused, path, and (sometimes) an end, and can be either steep or gentle. Most academic institutions like to think that their students experience the "gentle" variety...the long, excruciatingly slow, positive, linear slope. However, usually it is just long and excruciatingly slow.

Next is the simple curve, of which there are two basic types: convex and concave. With the "convex" curve, you quickly acquire a lot of information. "Whoa! Piece of cake. This is great! This program is really intuitive and easy to learn. I think I'll recommend it to all of my friends." However, soon you discover that you are bogged down in the Swamp of Despair, cursing yourself for even mentioning it to your dog and hoping against hope that all your friends were suddenly struck with amnesia so that they won't all launch a kamikaze attack in retaliation.

With the "concave" curve, on the other hand, you are lost in a thick fog for... like forever... not knowing where you're going, wondering why the heck you started this journey in the first place. The more intelligent among us bail out at this point and go back to playing Doom, but if you can hang on, gradually, the fog begins to lift. Then, like a greased pig down a laundry chute, the fog is swept away and you are left blinking in the sudden brightness and clarity.

Then there is the "on-the-fly" learning curve. This is usually the norm for me. Generally, this learning curve is extremely short-term, very specific (I only need one feature of a program for the particular project I'm working on), and intermittent – not unlike the dashed line on the highway.

A close relative of the "on-the-fly" curve is the "re-tooling" curve. This is the result of erratic, insidious, re-packaging of familiar territory that companies feel the need to do every so often. It is one of the great questions of human existence. When new versions of the software are released, why is it that all the tools you knew by heart have been randomly moved about or put into totally different menus? Do they think that we might not notice? I have a feeling that they know we enjoy learning and so they figure that they will give us a good time.

Most people are familiar with these types of curves, which are generally considered to be "positive," meaning you are increasing your knowledge. However, that isn't always the case. Less familiar are learning curves that are, at least in part, either flat (meaning that no matter how much time and effort you put in, you just can't learn anything), or negative, meaning you are losing it. At that point, it's time for some inspiring TV, like Jerry Springer, where at least you can regain some sense of intelligence by comparison.

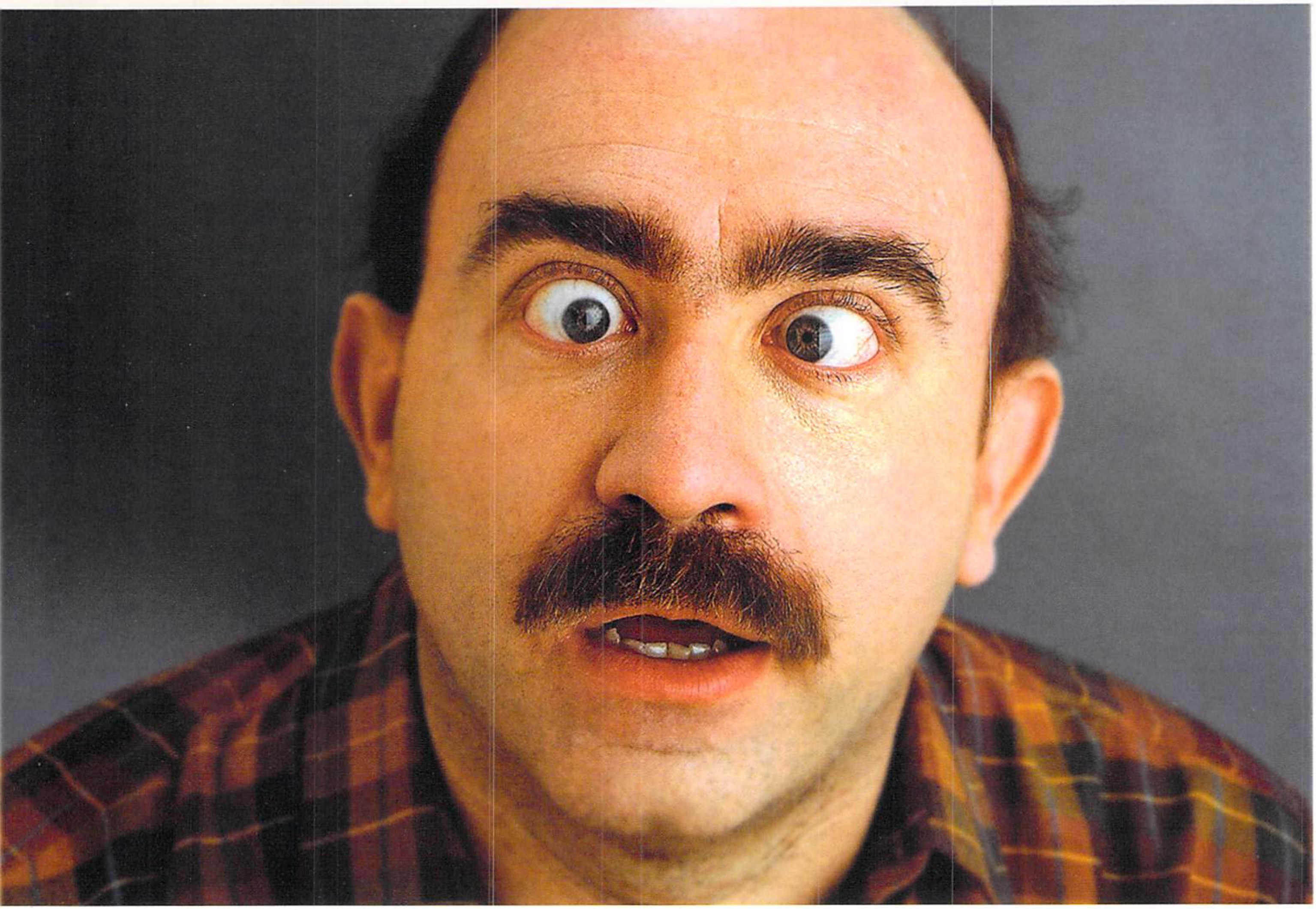
The most frustrating type is that of the Figure 8 learning curve. Not only are you fooled into thinking you are making progress, but you are actually re-learning the same exact stuff over and over again, in a never ending loop.

With the Spiral curve, you eventually do succeed, even though you have to learn stuff again and again. Fortunately, you learn a little bit more each time you go over it, eventually making progress.

Related to the Spiral learning curve is the Spirographic learning curve. Named after the Spirograph toy, the type of learning described by this curve is characterized by multiple periods of learning and loss. However, rarely are you learning the exact same stuff over again, although it often overlaps with stuff you already learned. The resulting curve is an interesting, beautiful, and complex pattern.

The learning curve that strikes the most fear in my heart is the X-treme curve that I call The Great Leap. While not technically a curve (not even a steep slope), it is the mental equivalent of BASE jumping – where no intelligence or learning is involved, you just have to DO it. I have experienced the 3D equivalent of this adventure more than once. Just thinking about it gets my heart racing! Now that I think of it, I bet I would prefer BASE jumping! In fact, leaping into empty space from the top of a remote, rugged, desert cliff and plummeting toward a jumble of ragged boulders sounds infinitely more enjoyable than some of the learning curves I have had to endure!

BRAD AND ANDREA CARVEY HAVE BEEN DOING COMPUTER ANIMATIONS FOR A LONG TIME. IN 1969 BRAD USED AN ANALOG COMPUTER, WHICH WAS THE SIZE OF A CAR, TO PRODUCE HIS FIRST COMPUTER ANIMATION. ANDREA, AN ARCHEOLOGIST, PREFERS TO DO SCIENTIFIC ANIMATIONS; HER CREDITS INCLUDE PROGRAMS LIKE DISCOVERY CHANNEL'S *UNDERSTANDING CARS*. BRAD IS AN ELECTRICAL ENGINEER AND AN EMMY AWARD-WINNING MEMBER OF THE VIDEO TOASTER DEVELOPMENT TEAM. HE PREFERS TO DO FEATURE FILM WORK. HIS CREDITS INCLUDE FILMS LIKE *MEN IN BLACK, STUART LITTLE, BLACK HAWK DOWN, KATE & LEOPOLD* AND *MASTER OF DISGUISE*.



INSIDE
LightWave [6]
INSIDE
LightWave 7
INSIDE
LightWave [8]

GET FOCUSED!

with the

High Quality. DVD-ROM. 60+ hours.

3Dgarage.com

3D | GARAGE™

LightWave® 3D Courseware

from Dan Ablan

Training from the author!

From NASA to the US Army, to NBC and FOX, to hobbyists, people around the globe are *finally* understanding the power of LightWave® 3D. See for yourself why the 3D Garage™ courseware is rated the best training you can find.

Download free tutorials now at **3D GARAGE.com**

AGA DIGITAL

AGA DIGITAL STUDIOS, INC. 1305 Wiley Road Suite 125 Schaumburg, IL 60173
847.755.4488 p 847.755.6401 f www.agadigital.com | www.3dgarage.com

Others promise. We deliver.

"IBM & AMD,
A relationship that delivers the
ultimate solution for
power hungry pipelines."
IntelliStation A Pro
AMD
64
Opteron
ibm.com/intellistation
IBM
ALEX & JUSTIN APPEAR COURTESY OF LUMINETIK

All IBM product names are trademarks or registered trademarks of International Business Machines Corporation.